TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Humanoid Robot Head Continuous Calibration for Active Stereo Vision, using Non-Linear Filtering Techniques

Nuno Miguel Banheiro Moutinho

Supervisor: Doctor Alexandre José Malheiro Bernardino
Co-Supervisor: Doctor José António da Cruz Pinto Gaspar

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with distinction

2017

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Humanoid Robot Head Continuous Calibration for Active Stereo Vision, using Non-Linear Filtering Techniques

Nuno Miguel Banheiro Moutinho

Supervisor: Doctor Alexandre José Malheiro Bernardino
Co-Supervisor: Doctor José António da Cruz Pinto Gaspar

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with distinction

Jury

Chairperson: Doctor José Alberto Rosado dos Santos Victor, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:
Doctor Antonios Gasteratos, Production and Management Engineering School, Democritus University of Thrace
Doctor José Alberto Rosado dos Santos Victor, Instituto Superior Técnico, Universidade de Lisboa
Doctor Alexandre José Malheiro Bernardino, Instituto Superior Técnico, Universidade de Lisboa
Doctor Jorge Nuno de Almeida e Sousa Almada Lobo, Faculdade Ciências e Tecnologia, Universidade de Coimbra
Doctor João Pedro Castilho Pereira Santos Gomes, Instituto Superior Técnico, Universidade de Lisboa

2017

Abstract

Humanoid robots are expected to, one day, fully replace humans in many hard and complex tasks. Nowadays, we can see robots being used to enhance the human capabilities, such as medical surgeons who use robotic platforms equipped with grippers to increase their precision during complex surgeries. This replacement however, foresees a future where robots will adapt to a human world and not the other way around. Complex tasks require complex robotic platforms and this level of complexity has been increasing through the years in terms of the number of degrees of freedom (DOF), the advance in the used sensors and the artificial intelligence level involved in the execution of their tasks. Unfortunately most robotic platforms are not "plug-and-play" in the sense they always need a calibration before executing a certain task, a process that could be quite challenging. There are two well known problems related to the calibration of humanoid robotic platforms that have been studied for years: the kinematic calibration and the stereo calibration.

The kinematic model of a robot provides, at every instant, the position and orientation of the robot's sensors from the angular measurements given by the motor encoders. However, a transverse limitation to most humanoid robots consists in using relative encoders in their joints instead of absolute ones. These encoders fix their zero value at the position they are turned on thus leading to an erroneous state of the robot's pose, if not properly initialized. A calibration is always required at start up to find the correct offsets of the joints that lead to an accurate kinematic model. Stereo vision, present in most of the robotic platforms, is used to give depth information and perform 3D reconstruction of the environment. It is extremely important in many tasks such as navigation or object manipulation. However, stereo systems are extremely sensitive to any small misalignments that may occur. Usually these systems are calibrated once, before performing a task and hopefully, maintain their calibration during operation. If the cameras are moved, calibration is lost and the system has to be re-calibrated.

In this thesis we developed two methods for online sensor based calibration that aim

to solve the two previously mentioned problems. We combine information from different embedded sensors such as stereo vision (cameras), inertial sensors (IMU) and relative encoders, with non-linear filtering techniques to completely calibrate a humanoid robot head in a real-time fashion. The proposed methods calibrate entirely the kinematic model from the base of the head to the robot eyes and provides an online and accurate stereo calibration to work with active vision. Several experiments were performed, first in simulation to validate the proposed methodologies and then in real case scenarios to prove the robustness of the calibration systems when dealing with real information from real sensors.

Keywords: calibration, humanoid robots, internal model, kinematic model, stereo vision, non-linear filtering

Resumo

Espera-se que um dia, os robôs humanoides substituam os humanos em tarefas difíceis e complexas. Hoje em dia, é possivel ver plataformas robóticas a ser usadas a nível médico para aumentar a precisão na realização de cirurgias ou em linhas de montagem para permitir uma rápida produção de forma praticamente ininterrupta. Esta substituição no entanto, prevê um futuro em que os robôs se irão adaptar ao mundo humano e não o contrário. Tarefas complexas requerem plataformas robóticas complexas e este nível de complexidade tem vindo a aumentar ao longo dos anos em termos do número de graus de liberdade, do avanço no tipo de sensores usados e no nível de inteligencia artificial envolvido na execução dessas mesmas tarefas. Infelizmente a maioria das plataformas robóticas não são do tipo "plug-and-play" no sentido que necessitam sempre de uma calibração antes de executar certas tarefas, um processo que pode ser bastante desafiante. Existem dois problemas bem conhecidos de calibração associados a robots humanoides que tem vindo a ser explorados ao longo dos últimos anos: a calibração ao nível da cadeia cinemática e a calibração do sistema de visão estereo.

O modelo cinemático de um robot permite perceber, em cada instante, a posição e orientação de cada um dos sensores que compõem a plataforma robótica, através de mediçoes angulares obtidas pelos sensores de rotação (encoders) aplicados a cada junta. Contudo, uma limitação comum à maior parte dos robots humanoides consiste em usar encoders relativos nas juntas dos seus motores, em vez de encoders absolutos. Estes sensores fixam o seu zero na posição na qual são inicializados, o que leva à constante existencia de offsets nas juntas e por conseguinte, a um estado errado da pose do robot. É, por isso, sempre necessária uma calibração inicial por forma a encontrar os valores de offsets das juntas que definem o zero absoluto de cada uma. A visão estereo presente em grande parte das plataformas robóticas permite ter uma percepção de profundidade e reconstruir tridimensionalmente um cenário. Isto é extremamente importante em determinadas tarefas tais como navegação ou manipulação de objectos. Contudo, o sistema de visão estereo é extremamente sensivel a pequenos desalinhamentos que possam existir,

o que origina uma percepção de profundidade errada. Normalmente estes sistemas são calibrados uma única vez, antes de realizar uma determinada tarefa, onde se espera que a calibração se mantenha durante toda a operação. No entanto, assim que as camaras se movem, perde-se a calibração, e todo o processo terá que ser realizado novamente.

Nesta tese desenvolvemos dois métodos para calibração online, usando informação sensorial proveniente da plataforma robótica, que visam resolver os dois problemas acima mencionados. Combinámos informação de diferentes sensores tais como visão stereo, sensor inercial e "encoders" relativos com técnicas de filtragem não linear para calibrar totalmente, em tempo real, a cabeça de um robô humanoide. Os métodos propostos calibram toda a cadeia cinemática desde a base da cabeça até aos olhos do robô, calibrando também de forma online e precisa a visão stereo, por forma a ser usada com visão ativa. Foram realizadas várias experiências, primeiro em simulação para validar os métodos propostos e posteriormente num cenário real para comprovar a robustez do sistema de calibração quando confrontado com informação proveniente de sensores reais.

Palavras chaves: calibração, robots humanóides, modelo interno, cadeia cinemática, visão estereo, filtragem não-linear

Agradecimentos

Gostaria de agradecer aos meus orientadores, Professor Alexandre Bernardino pela simpatia, amizade e disponibilidade para discutir qualquer assunto sempre com abertura, dando inputs valiosos (e por ser Sportinguista), e Professor José Gaspar, pelas longas discussões e conselhos úteis que ainda hoje sigo, e pela sua atenção ao pormenor que no final, acaba sempre por fazer a diferença. Um agradecimento ao Professor José Santos-Victor por ser o comandante do VisLab e por ser um exemplo de como se deve gerir um laboratório: com amizade, confiança e sentido de humor apurado (apesar da má escolha futebolística).

Gostaria também de agradecer a todos os amigos que fiz durante os meus anos no laboratório. Ao Ricardo, Manel, Jonas, Matteo, Giovanni, Rui, Ricardo Nunes, Ana, Pedro, Plinio, Athira e a todos os outros que de uma forma ou outra privaram comigo e me deram o prazer da vossa companhia.

Queria também agradecer a todos os meus amigos que sempre me apoiaram nesta longa caminhada. Ao João e Mafs, Fred e Ana, Xixo e Maria (e pequena Teresinha), Caria e Joana, Bernas e Rita (e pequena Madalena), Gonças, João Nuno, Uish e Margarida (e pequeno Pedrinho), Fonzie, Vasco, Paulo, David, André, Fernandes, Bastos e Arnaldo (e a pequena Di), Maffy, Guida e a todos os que fazem parte da minha vida. A vossa amizada significa muito para mim e a vida não seria de certeza a mesma sem voces.

Queria agradecer ao meu irmão, aos meus avós e principalmente aos meus pais, por me terem apoiado incondicionalmente em todas as decisões que tomei. Foi graças a vocês que consegui chegar onde cheguei. Obrigado por estarem sempre presentes e principalmente pelo vosso apoio.

Por último queria agradecer à minha Ana, o meu porto de abrigo, minha companheira, conselheira e amiga, que me apoia em todos os momentos, nos bons e nos menos bons. Prometo compensar-te por todos os fins de semana perdidos em que não pudemos namorar ou viajar por ter que ficar a trabalhar na tese. Obrigado por seres quem és.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 A Brief History of Robotics

The term robot, or robota from the Czech meaning servitude, was first applied in the fictional play R.U.R. (Rossum's Universal Robots) in 1921 and replaced the popular use of the word automaton, describing a self-operating machine. From that moment, the idea of having robots among us, completely integrated in our society began to grow, even though concepts akin to a robot can be found as long ago as the $4th$ century BC. Many writers started including robots in their narratives and the science fiction genre gained a new impetus by challenging people's imagination: people started thinking of a fictional and futuristic society where life could be enjoyed while robots do all the hard and boring work. In 1942, in an attempt of setting basic rules to assure robots could securely co-exist among us, the science fiction writer Isaac Asimov formulated the Three Laws of Robotics which can still be applied today to many robotic platforms.

70 years have passed and our human society is now completely dependent on robotic platforms to perform complex tasks in order to fulfill the requirements imposed by the industry. The automotive industry for example, uses complex robotic platforms for years to assemble vehicles in a better and faster way compared to humans. Robotic platforms can perform repetitive tasks without getting tired, bored or hungry. They can work 24 hours a day, 7 days a week until they brake or get obsolete, moment when they are rapidly replaced by other. The smartphones "boom" may not have happened if it wasn't the integration of robotic platforms in the assembly process, which requires extremely high precision mechanisms, difficult to perform by any human hand. There are robotic platforms already being developed to replace us in our everyday tasks such as ironing, cleaning or cooking. Robots are being used to enhance the human capabilities, such

as medical surgeons who use robotic platforms equipped with grippers to increase their precision during complex surgeries. Robotic platforms do not usually take a human form even though these already exist. There are humanoid and non-humanoid robotic platforms already being developed for military purposes that may replace humans in war fields. The need to perform increasingly complex tasks more quickly and accurately was the starting point for a continuously growing and extremely important parallel industry: the robotic industry.

1.2 The Internal Model in Humans and Robots

Humanoid robots are expected to, one day, fully replace humans in many hard and complex tasks. This replacement however foresees a future where robots will adapt to a human world and not the other way around. Complex tasks require complex robotic platforms and this level of complexity have been increasing through the years in terms of the number of degrees of freedom (DOF), the advance in the used sensors and the artificial intelligence level involved in the execution of its tasks. Unfortunately most robotic platforms are not "plug-and-play" in the sense they always need a calibration before executing a certain task, a process that could be quite challenging when working with very complex platforms. Like humans, robots must use their sensors to know their internal model state. The authors in [48, 28] discuss the possibility of the human cerebellum containing an internal model of the entire body that can predict consequences of actions based on sensory information. It can also do the inverse process where, given muscles information the (inverse) internal model predicts information from the sensors. Such process requires a training phase where the model is learned and continuously compensates position errors in the muscles using feedback from its sensors. Once learned these models are used to perform muscle control and, as claimed by the authors in [48, 28], can replace feedback control which is too slow for everyday tasks.

In [22] the authors study the body representation in humans and how is the body structured in our brain. This representation may not be unique and presents, as the authors mention, plasticity properties meaning the body is always updating its representation to better adapt to changes that may or will happen. The learning process starts in a very early stage when infants begin to interact with their own body and with the environment. This interaction allows them to learn their internal model, referred by the authors in [22] as the visual-proprioceptive calibration of the body. The authors then extend the concept of body schema to the robots and expose the benefits of having such models for this kind of platforms for control purposes or to predict actions based

on motor commands only.

An internal model is extremely important when dealing with multi-DOF platforms and only a full representation of their kinematics allows its correct operation. This extension of internal model representations in robots can therefore work as a tool to verify hypothesis of the human internal model. Humanoid robots are tested under several conditions and their behaviour is compared to the ones presented by humans. A wide variety of tests is performed, to analyse their adaptability to sudden changes in their body, their capability to use tools or to interact with the surroundings using proprioception and their learned and calibrated internal model. These tests will allow the validation or rejection of previously formulated hypothesis with a direct observation in human-like robotic platforms.

In [47] the authors review different learning methods for the internal model, specially focusing on three ways the system can learn from its interaction with the environment: from supervised learning, reinforcement learning and unsupervised learning. In supervised learning the environment provides for each input an explicit output, and the goal is to find a mapping between the inputs and the outputs. The system will then try to minimize the error between the real and predicted signals. In reinforcement learning for each input to and output from the system the environment gives feedback in terms of reward or punishment. The learning system will then try to maximize the sum of all possible rewards given by the environment. Finally in unsupervised learning the environment provides an input but gives no output or feedback in terms of punishment or reward. The calibration system presented in our thesis falls in the first group, with a self-supervision of the outputs provided by the environment. The system will therefore try to minimize the errors between the observed outputs and those that were predicted using the learned internal model of the robot head. Another interesting point that was raised by the authors of [47] is the existence of multiple internal models called basis models that can be combined to interact with elements of the environment that were never seen before. If you know how to ride the roller skates, you may be able to ride ice skates too, even though they are different. Humans keep these models in memory, where the cerebellum is responsible for the long-term maintenance of the internal models. Trying to control each muscle of the human body is much harder than just keep some basic primitive patterns of muscle configurations that can be combined to achieve the same goal.

The ability to predict sensorial information based on proprioception and on our internal model was studied by Berthoz in [2]. He claimed humans use their internal models and information from their muscles to predict the consequences of their actions.

If I pick up a pen and hold it in my hand, I can close my eyes and track it even though I am not directly seeing it. I always know its position and orientation in space because it is in my hand and I know where my hand is. When I finally open my eyes to confirm my prediction, the error between the real and predicted state of the pen is very low since we have a tremendously accurate internal model of our body. This concept of predicting the consequences of our own actions was further applied to robotics when the authors of [8] proposed an anticipatory architecture where a robotic platform could predict the consequences of its own actions using its learned internal model. This concept was denoted Expected Perception (EP) and then exploited in several other works [25].

The authors in [8] proposed an antecipation-based perception-action scheme for robots, using the EP concept, where virtual images from the environment were generated using both proprioception and past information from the scene, and were further compared with the real images to detect unexpected events. In order to interact with a dynamic environment a robot must be able to distinguish the independent elements that can move freely from all the others whose dynamics only depend on its own (static world objects, agent's body parts, objects manipulated by the agent). Humans tend to create models of the environment using past information and all the experience learned during lifetime. We know that if someone is walking in our direction at a certain speed, our paths will eventually cross at a certain time instant that we can predict, given our model of movement extrapolated to another person. The assumption of a static world is extremely helpful to perceive movement that was caused by us: if I move my head to the left I expect to see everything moving to the right thus obeying to my model of movement. Any captured movement not obeying to the expected behaviour is considered unexpected and captures our attention. If someone throws me a ball while I am moving my head, I will detect it and try to react in order to avoid it. This is only possible due to an extremely well calibrated internal model.

In [30] we took this concept a little bit further and applied the EP concept to 3D reconstruction, where 3D information from the environment was only updated if there was a difference between the real sensorial information provided by the cameras and the one predicted by our internal model. The selective update of the 3D information, instead of a full and continuous update, highly reduces the computational burden of the system during tasks execution. In the published work we show results with the system detecting an unexpected event (hand appearing in front of the camera) during rotation of the robot head, while predicting all the other events in the world. In all these applications the internal model plays an important role and is of utmost importance to have a calibrated model of the platform during operation.

The EP concept was further used as the basis of an international project named RoboSoM, where the sense of movement proposed by Berthoz in [2] was applied to a walking humanoid robot. The sense of movement is the ability the human brain has to perceive the environment and predict the consequences of actions before activating the muscles to perform a certain movement. The author of [2] claims we base our control in brain predictions instead of sensory feedback, with the brain making decisions in a fraction of the time it would take if no predictions were used. The idea that we can predict the consequences of our own actions using our internal model was extrapolated to robots in that project. The robot's internal model provided accurate information about its state at every time instant allowing it to predict certain events and adapt to sudden changes of the environment during walking while tracking an object. By using calibrated proprioception the robot was able to predict the position of the tracked object even when it was not visible which is a common human behaviour. That project confirmed some theories of how humans perceive the world and use their internal model to better interact with the surroundings.

An internal model, if properly calibrated, is fundamental to predict the consequences of the platform's own movements in a certain physical scenario. This predictive mechanism has advantages since you can base its control on predictions instead of real sensor readings. However, an accurate calibration status can be highly time consuming and prone to errors, even if manually performed by experts. An alternative way to handle such complex systems is to exploit the robot's embedded sensors to design automated calibration methods which are faster, safer and more accurate than manual tuning.

1.3 Objectives and Contributions of the Thesis

Every robotic platform requires a calibration process and in this thesis we will focus on the specific case of humanoid robot heads which are usually equipped with stereo vision (cameras), inertial sensors (IMU) and absolute or relative encoders that provide angular measurements of the motor joints. Comparing a robotic platform with a human being, we can make a parallel between their sensors. The stereo vision corresponds to the human eyes, responsible for the human vision. Considering they are located at the last link of the head's kinematic model, they can easily perceive any movement performed by the head. Any rotation will be mapped to the cameras' images, reason why these sensors are very useful for calibration purposes. The IMU corresponds to the human vestibular system responsible for the human balance and perception of linear accelerations and angular velocities. Often, this sensor is located on the top of the head,

thus it can only perceive movements from the head excluding the eyes and can be used for head stabilization. The joint encoders correspond to the human muscles between each joint which provide angle measurements from the muscles tension. Considering that the human body knows its kinematic model it can determine the position and orientation of each sensor. A perfectly calibrated internal model consists of a system where each sensor is able to predict measurements of other sensors, e.g using joints trajectories it's possible to predict the linear accelerations measured by the IMU (and vice-versa). However an accurate calibration result is always difficult to obtain since it depends on the quality of the sensors, on the accuracy of mechanical parts, or mounting errors of the cameras that are not measurable using visual information only. It is only when we combine different sensor measurements that the problem becomes fully observable and a complete calibration can be achieved.

With this thesis we aim at combining information from the different embedded sensors and non-linear filtering techniques to completely calibrate a humanoid robot head in a real-time fashion. The proposed method calibrates entirely the kinematic model from the base of the head to the robot eyes and provides an online and accurate stereo calibration of extrinsic parameters to work with active vision. The stereo calibration, in its typical configuration (horizontal baseline passing on the optical centers and orthogonal to both optical axes) has inherent ambiguities in the map from image features to extrinsic calibration parameters. For instance, points at infinity have zero image motion for any translation of one camera with respect to the other; a 3D point aligned with the camera optical axis also has zero motion independent of the camera roll rotation; horizontal translations and rotations around the vertical axis can generate an identical motion of points projected in the center of the image; idem for vertical translations and tilt rotations. Noticing that the observability problems are related to the depth and the location of the point projection in the image, and are different for each calibration parameter, we propose to filter out the problematic points from the estimation of the corresponding parameter, at each time step. Despite state-of-the-art parameter estimation methods (Kalman-like filters) also use the observation sensitivity to reduce the weights of observations in the parameter update phase, these weights only asymptotically go to zero as the covariance of the parameters grow to infinity, resulting in parameter drift and a large sensitivity to noise. Our proposed approach drops out the points that are unable to provide sufficient information (enough signal-to-noise ratio) for the estimation of a given parameter, so it is able to reduce estimation variance and prevent drifts. We present an observability analysis for the stereo calibration problem that provides guidelines for the selection of which measurements are informative enough

to estimate each parameter of the stereo system, with respect to the present noise levels or pixel quantization errors. We apply this study to the design and implementation of an online stereo calibration system. Both calibration methods do not require artificial patterns, work online, converge quickly to the solution given enough informative points, and do not diverge in the presence of non-informative points.

1.3.1 Main Contributions

In summary, the main contributions of this thesis are:

- use of an Implicit Extended Kalman Filter to solve complex kinematic and stereo calibration problems applied to robots;
- feature selection based on an observability analysis to improve the results of the stereo calibration filter;
- design of easy-to-use calibration systems that use information from embedded sensors and natural information from the environment, without the need of any markers, special calibration patterns or complex calibration procedures;
- implementation and testing of the calibration systems in real robotic platforms with an exhaustive validation of results showing high precision and fast convergence rates;

1.3.2 Publications

Excerpts of this document were based in previous works developed during our thesis, namely:

- *"An expected perception architecture using visual 3D reconstruction for a humanoid robot"*, [30], based on the EP concept applied to real-time 3D reconstruction;
- *"Online calibration of a humanoid robot head from relative encoders, IMU readings and visual data"*, [29], with the proposed head calibration system;
- *"Markerless online stereo calibration for a humanoid robot"*, [31], where we proposed a kinematic based solution to the stereo calibration problem;
- *"Good Features for On-line Stereo Calibration of Active Vision Systems"* - Nuno Moutinho and Alexandre Bernardino, submitted to IEEE Transactions on Robotics Journal, 2017 - where we present the proposed solution for the stereo calibration problem with the observability analysis

1.4 Thesis Outline

The thesis is organized as follows:

- Chapter 2 describes the head and stereo calibration problems in detail by stating the main challenges of each one. A review of the major kinematic and stereo calibration techniques is also presented with a complete analysis of their strengths and weaknesses to deal with these problems.
- Chapter 3 presents all the mathematical and computer vision tools used in our work, for a clearer understanding of the thesis. The presented tools will be referenced throughout the thesis.
- Chapter 4 presents our real-time solution to the head calibration problem, with a full explanation of the system's architecture and implementation.
- Chapter 5 provides an extensive experimental evaluation of the head calibration system in terms of accuracy and repeatability, with a validation of the proposed solution in a simulated environment and the application to a real robotic platform in real conditions.
- Chapter 6 presents our real-time solution to the stereo calibration problem with a complete observability analysis of the stereo calibration problem and how this analysis can improve the stereo calibration results.
- Chapter 7 provides an extensive experimental evaluation of the stereo calibration system in terms of accuracy and repeatability, with a validation of the proposed solution in a simulated environment and the application to a real robotic platform in real conditions. Finally we show how the two calibration systems can be combined to fully calibrate the robot's internal model.
- Finally chapter 8 concludes this thesis, presenting a summary of our main contributions to accurate robotic calibration.

Chapter 2

Problem Formulation and Related Work

Many robotic tasks are based in object manipulation where a precise depth perception at close distances is important. Humans rely on stereo vision to get this information within a manipulation region and constantly move their eyes to get the information they need. A humanoid robot equipped with two cameras can, in principle obtain the same information under the same conditions. But what will happen if the robot does fast eye movements the way humans do? Can the robot rapidly adapt its stereo vision to these changes? A human can also move its head and look around, creating a temporary memory of the surroundings. Can a robot do the same?

Humans have advanced perceptual skills that provide real time information about the full state of their body. In robotic language, humans have an accurate calibrated kinematic model that is constantly adapting to changes in their own body or in the environment. The same must happen for humanoid robots. There should always be running an online calibration process to ensure the robot's internal model is constantly calibrated no matter the tasks it is performing.

In this thesis we propose a solution to fully calibrate a generic humanoid robot's head, from neck to eyes, equipped with stereo vision and motors in their joints. This solution can be extended to other humanoid robot heads equipped with the same type of sensors. The calibration problem is different when dealing with the head motors or the stereo vision and we will separately explain in detail each of the problems we are solving.

2.1 Head Calibration Problem

A transverse limitation to most humanoid robots consists in using relative encoders in their joints instead of absolute ones. These encoders fix their zero value at the position they are turned on thus leading to an erroneous state of the robot's pose, if not properly initialized. A calibration is always required at start up to find the correct offsets of the joints that lead to an accurate kinematic model. A well calibrated kinematic model can predict the correct position of each sensor (joint, IMU, cameras), represented in a platform centered reference frame, by just applying the correct kinematic transformation using solely the encoders measurements as input.

Using absolute encoders in the joints may seem as solution for this specific problem. However, it is not guaranteed that the zero position of each joint matches the absolute zero of the kinematic model, mostly because many experimental platforms are hand mounted. Misalignment due to mounting errors of the sensors need to be detected and corrected before the platform is used otherwise there will be undesirable offsets in the position of the end-effector when performing the tasks it was designed for (e.g. the robot hand grasping an object). Having misaligned absolute encoders in the motor joints is as bad as having relative encoders, except that for the former the offsets are always constant requiring only a single calibration, while the later have to be calibrated every time the motors are switched on.

Assuming the robot has absolute encoders that were perfectly mounted in the motor joints, the continuous use of the robotic platform generates drift in some of the joints, that can not be detected solely using information from the encoders, mainly on those supporting most of the platform's weight. Common robotic heads present relative encoders, misaligned absolute encoders and slow drifts in mechanical positions due to wear and impacts. In this thesis we will develop a solution to estimate a precise kinematic structure of the system despite these problems, at a software level.

2.1.1 Mathematical Formulation

The robotic platform is represented by a serial kinematic chain which consists of multiple rotation joints serially coupled, as seen in figure 2.1a).

Each joint i can rotate by an angle θ_i around its axis of rotation. In a calibrated internal model, θ_i corresponds to the exact measurement given by the encoder sensor, e_i. However if the internal model is not properly calibrated, there is an offset δ_i that must be considered and the angle θ_i becomes a linear combination of the encoder measurement and the offset, as seen in figure 2.1b), with the relation given by:

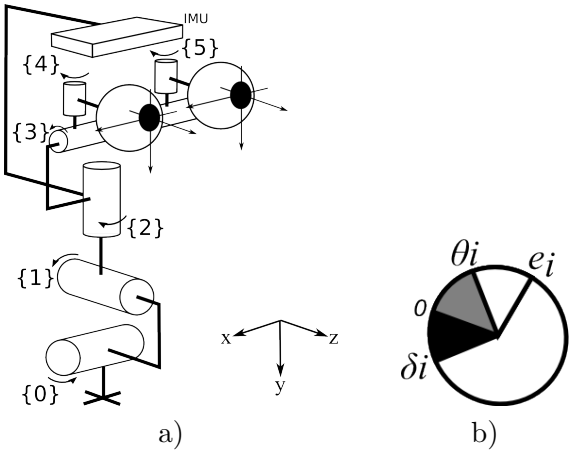


Figure 2.1: The head kinematic model; a) the corresponding serial chain of joints, from the neck base $\{0\}$ to the eyes final joints $\{4\}$ and $\{5\}$; b) a cross section of a joint $\{i\}$ showing the relation between the encoder measurement e_i and the offset δ_i in the calibrated joint angle θ_i

$$\theta_i = e_i - \delta_i \tag{2.1}$$

Let ${}^{i+1}T_i\left(\theta_i\right)$ represent a roto-translation between two consecutive joints, i and $i+1$. This roto-translation is a function of the rotation angle θ_i associated to the ith joint. The complete roto-translation from a reference frame $\{n\}$ to the base of the kinematic chain $\{0\}$ is written as

$${}^{n}T_0\left(\theta_0, \ldots, \theta_{n-1}\right) = {}^{n}T_{n-1}\left(\theta_{n-1}\right) \ldots {}^{2}T_1\left(\theta_1\right) {}^{1}T_0\left(\theta_0\right) \tag{2.2}$$

We can easily see that an offset in one of the primary joints or the composition of multiple offsets through the kinematic chain introduces large errors in the final roto-translation of its end-effector. The goal of our work is to estimate all the offsets δ_i so the roto-translation given by the kinematic model adequately reflect the real state of the system and the relation (2.1) holds for every considered joint.

In the case of a typical robotic head equipped with an IMU and stereo cameras, as the one in figure 2.1a), we want to find the offsets defining the head's absolute zero position, with both cameras pointing to the front (projection planes orthogonal to the floor) and a gravity vector reading given by the IMU corresponding to a vertical vector pointing down. Thus, the vector containing all the offsets to estimate is then represented as

$$x_H = \begin{bmatrix} \delta_0 & \dots & \delta_5 \end{bmatrix} \tag{2.3}$$

2.1.2 Related Work

Some works have addressed the robot self-calibration problem via an non-linear parameter estimation problem given sufficient input data from the robot sensory system. The Body Schema (a denomination for the set of kinematic parameters that determine the robot's model – not only joints offsets but also link lengths and angles) have been estimated with local optimization methods given appropriate initializations.

In [20] the authors present an online learning system for the body schema of the Hoap3 robot, a humanoid robotic platform with 24 degrees of freedom (DOF). The system uses information from the propriosensors and from stereo vision to correct and calibrate its internal model, by tracking its end-effectors using color markers. Although the authors show good results in simulation, for the considered approach, they recognize it is difficult to implement such a solution in a real robot with many DOF since it would take a large amount of time for the robot to explore all its joints space using random movements. In particular, for some of its end-effectors the acquired information may be insufficient due to lack of direct visibility (the robot may not be able to always observe its feet). The random exploration of the whole joints space is widely time-consuming and may not provide enough information for the system to correctly converge. An intelligent exploration of this space could highly improve the calibration results while reducing the total calibration time.

The authors in [6] present an online active learning algorithm for the body schema of a robot. However, the main contribution of that work resides in the way the robot explores its joints space. This exploration, using active learning techniques, is not random but selective in the acquired observations and executed movements. The robot will only get observations that could actually improve its calibration and explore areas where uncertainty remains high. The accuracy and speed improvement by using this technique is considerable when comparing with a random approach. It is of utmost importance to understand if an observation will actually add useful information to the system or simply act as noise.

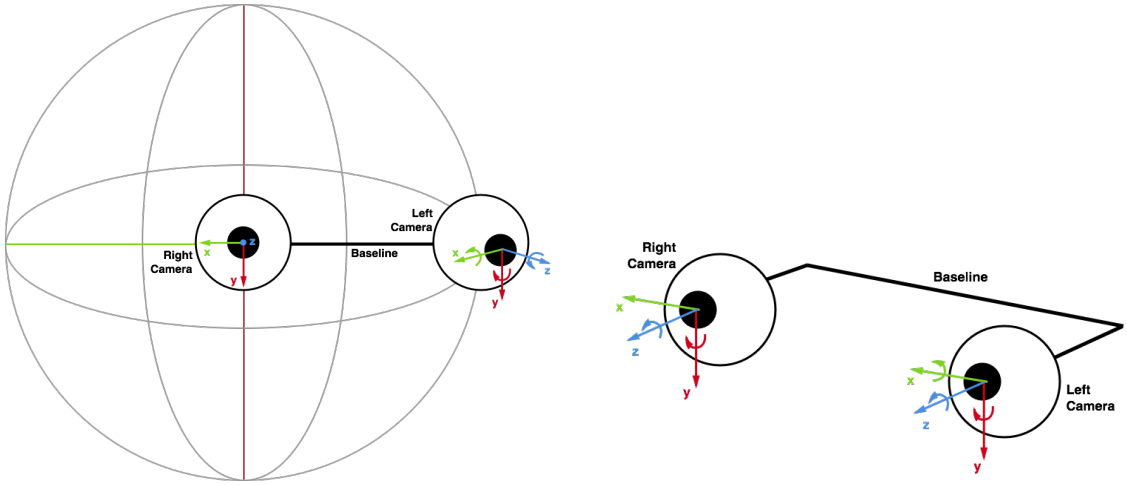
In the work described in [42] the authors present a kinematic calibration system for a pan-tilt structure with a camera at its end-effector. The system estimates the homographies between consecutive time-instants by tracking points on the images. This method allows the calibration of the joint angles offsets that were present at startup

in the relative encoders. Robotic platforms with relative encoders require a kinematic calibration before any operation and in these cases, calibration time is crucial. The authors in [38] present a kinematic calibration algorithm for a robotic head, similar to the one used in our thesis. This robotic head is equipped with relative encoders and they propose a solution to estimate the offsets of each joint, from the neck base to each of the eyes. The problem is separated into two sub-problems that are solved using different methods, more suitable for each case. To calibrate the head the authors use information from the inertial sensor in order to find the joints offsets that enables the alignment of the head with the robot's body. To calibrate the eyes the authors used the same approach as in [42]. None of the implemented methods takes noise into consideration which may lead to erroneous estimations. Moreover, these approaches were not designed to be used in an online manner and can not respond or adapt to changes that may occur during operation. Due to heavy use these platforms usually have slow mechanical drifts in their motor shafts due to wear, impact and strain that deform the parts, where the encoders can not provide any measurements. An online calibration system can detect these changes and adapt its estimates along time as we will show in this thesis.

2.2 Stereo Calibration Problem

In the head calibration problem we assumed calibrated cameras. Here we study how this can be achieved using only the images from the robot's cameras. The stereo roto-translation obtained from the kinematic model hardly describes the real roto-translation between the two cameras since it assumes both sensors are perfectly mounted, which is not true in general.

Stereo vision systems are extremely sensitive to any small misalignments that may occur. Usually these systems are calibrated once, before performing a task and hopefully, maintain their calibration during operation, e.g. assuming the cameras do not move. If the cameras are moved, calibration is lost and the system has to be re-calibrated. Even though some robotic head platforms are equipped with encoders in their motor joints, in many cases their precision is not enough to maintain a good calibration. Also, mechanical artifacts (e.g backlash, static friction) produce control errors that are hard to model. The sensitiveness of stereo reconstruction to errors in the relative pose of the cameras makes this a hard problem to solve specially when dealing with active vision. Humanoid robots, like humans, can perform two types of eyes movements: version and vergence. The stereo transformation is highly dependent on the movement that was applied to the robot's eyes. The kinematic model could, in part, explain this transformation if the



a) b)

Figure 2.2: Spherical model centered in one camera (right) and having the other camera (left) moving along the sphere surface; a) global model representation; b) the stereo model parameters

two cameras were perfectly mounted in their motor joints, which is not often the case. There are always misalignements due to mounting errors of the cameras that can not be explained by the kinematic model. An online calibration is required to continuously provide the correct transformation between the two cameras.

Stereo systems with a fixed baseline are common to many humanoid robots. This can be modeled by a 5 DOF representation, having one camera fixed and centered in a sphere and the other arbitrarily located along the surface of that sphere, keeping the distance to the center always constant, as seen in figure 2.2a). The calibration problem is then formulated as estimating the parameters of the rigid transformation of the second camera with respect to the fixed one, represented by a rotation matrix ${}^{R}R_{L}$ and a translation vector ${}^{R}t_{L}$. We adopt a parametrization for the rotation using X-Y-Z fixed angles [7] using angles r_x for camera pitch, r_y for camera yaw and r_z for camera roll, as seen in figure 2.2b):

$$ {}^{R}R_{L} = \begin{bmatrix} \mathrm{c}_z & -\mathrm{s}_z & 0 \\ \mathrm{s}_z & \mathrm{c}_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{c}_y & 0 & \mathrm{s}_y \\ 0 & 1 & 0 \\ -\mathrm{s}_y & 0 & \mathrm{c}_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}_x & -\mathrm{s}_x \\ 0 & \mathrm{s}_x & \mathrm{c}_x \end{bmatrix} \tag{2.4} $$

where c_w and s_w are abbreviations for $\cos(r_w)$ and $\sin(r_w)$ respectively. The translation vector is parametrized by translations t_y and t_z. The third translational component t_x

comes directly from the condition of fixed baseline, $||^R t_L|| = B$.

$$^R t_L = \left[-\sqrt{B^2 - t_y^2 - t_z^2}, \quad t_y, \quad t_z \right]^T \tag{2.5}$$

Thus, the vector of parameters to estimate is:

$$x_S = [t_y, t_z, r_x, r_y, r_z] \tag{2.6}$$

Although mathematically it is possible to estimate all the parameters from only 8 image points pairs [18], the parameters are very sensitive to factors that directly influence the calibration: the image points used as measurements, the cameras resolution and the length of the baseline.

2.2.1 Related Work

The Bouguet Toolbox [3] is well known amongst the computer vision community and is still considered the state-of-the-art for camera calibration in terms of accuracy. This toolbox estimates both the intrinsic and extrinsic parameters of stereo cameras by using a chessboard pattern. However, it requires the acquisition of many calibration images, where the pattern is visible at different regions of the images, with different orientations, and the manual selection of image points belonging to the pattern, which is highly time consuming. In [19] a new calibration pattern is introduced, different from the usual chessboard patterns used in several calibration toolboxes [3, 4]. This pattern ensures the detection of many image features (SIFT [26]), from different points of view. The advantage is that it doesn't require full visibility in both images of the stereo pair, which is very useful for setups with a large baseline. However, this form of calibration can not be used for online applications where cameras move. It can be used, though, for static stereo platforms or as ground-truth to test other stereo calibration methods. Most systems relying on calibration patterns are highly time consuming and require human intervention. Their precision however is greater than any other calibration system reason why they are still in use.

In [15] it is proposed a method to calibrate a stereo platform with 5 degrees-of-freedom (DOF) and fixed baseline (distance between cameras), by minimizing the distance of image features to the epipolar lines. The extrinsic parameters are encoded in 3 angular DOF's for one camera and 2 angular DOF's for the other camera. The authors note observability problems since two of the rotations are coupled and only estimate their differential value, which was always correct when compared with ground-truth values.

Thus, the method calibrates only 3 DOF instead of 5 DOF.

In our first approach to the stereo calibration problem, [31], we introduced the concept of virtual joints to explain the mounting errors of the cameras in a general stereo kinematic model. The system calibrated the absolute zero of each joint (real and virtual) allowing movements of the eyes with a rapid adaptation of the calibration system to those sudden changes. The system was able to provide a stereo calibration by minimizing the epipolar constraint error between the two uncalibrated cameras. However, the kinematic representation used in this work only had 4 DOF resulting in an approximated estimation of the real roto-translation given the incomplete model of the eyes. Allowing both eyes to freely rotate around their 3 axis showed to be a problem due to observability issues which resulted in a system's divergent estimates, specially when using points at large distances.

[24] developed a calibration system for a multiple camera rig mounted on top of an autonomous vehicle, based on an Extended Kalman Filter (EKF [36]). They exploit prior knowledge about the environment and use the ground constraint and plane induced homographies between consecutive frames to calibrate the extrinsic parameters of the structure. However, the method is specific for the particular application considered. The authors recognize the limitations of their approach and present some cases where it fails. Another calibration algorithm for a fixed stereo rig is presented in [34]. The authors simplified the calibration problem by removing the translational parameters and by using an approximation for small rotations between the two cameras, valid for the case where the two cameras are almost in a parallel configuration. Any perturbation to this configuration may not be correctly represented by the system thus resulting in a wrong calibration.

In [12] the authors use a top down approach where a rough estimation of the stereo cameras parameters and the output of multi-view stereo cameras is used to refine the search for image matches that can improve the previous cameras calibration. They use a standard bundle adjustment algorithm to best fit the image data to the 3D points obtained from a rough reconstructions of the scene, using low resolution images to speed up the whole calibration process. This system calibrates multi-view images from widely separated cameras, which does not obey to the typical stereo model. This can in part explain the chosen techniques and the approach of using data from the 3D reconstruction to improve the cameras calibration, specially the orientation of the features from the reconstructed objects' surfaces which is different from camera to camera and provides valuable information of the multi-view geometry imposed by the scene.

In [44] is proposed an online stereo calibration system based on a modified bun-

dle adjustment algorithm. Good results are achieved in terms of the output of visual odometry experiments using the calibrated stereo parameters. However, because most of the points are at a large distance from the cameras, these results do not imply a good stereo calibration. In fact, we can clearly see from the calibration results that the system has lack of observability, specially for the translational parameters, resulting in considerable errors. These errors have no impact on the visual odometry output since the rotational parameters are sufficient to generate the desired results. When working with humanoid robotic platforms, using active and stereo vision in object manipulation tasks, the absolute position of the cameras is extremely important. Having large values in the translational parameters would affect the execution of the tasks and more accurate stereo calibration systems are required.

[46] introduces a stereo calibration solution for a robotic head with active vision that permits saccadic eye movements without losing stereo information. They find the constant roto-translation matrices from each eye pan joint to the corresponding camera reference frame, which describe the mounting errors of the cameras. Nonetheless, this calibration method requires the previous acquisition of extrinsic camera calibration data by observing a calibration pattern while rotating the head. This requirement is time consuming and not suitable for systems where operational conditions can change often.

[9] presents a hand-eye calibration system where the stereo calibration plays an important role. The system first calibrates the stereo cameras offline, using the 8 Point Algorithm [18]. The robot then performs predefined movements with its hand in front of the camera thus recovering the 3D position of its finger. The system will then minimize the error between the finger tip position given by the stereo reconstruction, with the one obtained using the robot's kinematic model. This calibration is intended for grasping tasks where this transformation (hand-eye) is extremely important. Although the authors present good grasping results, the used stereo calibration algorithm is known for being highly sensitive to noise in the image features, which is not considered during the calibration process. Even though this algorithm requires only 8 points to generate the Essential Matrix [18, 17], not all are equally good or provide enough information about the state of the stereo structure.

From the literature we can see that many solutions for the stereo calibration problem suffer from observability problems. There are ambiguities in the stereo parameters under estimation that may lead to erroneous estimates. Some of the approaches tried to attenuate the influence of low observability by simplifying the global problem at the cost of loosing degrees of freedom.

Chapter 3

Mathematical and Computer Vision Tools

In this chapter we are going to explain in detail all the tools used in the implementation of the system described in this thesis. These mathematical and vision tools play an important role in the implemented architecture and must be fully understood before going into details on the system's architecture.

3.1 Camera Model

The camera model used in this thesis is the pinhole model [17]. The pinhole camera model maps 3D camera coordinates to 2D homogeneous image coordinates, projected into the image plane. This projection is obtained using the intrinsic matrix K, [17].

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Each parameter corresponds to a certain geometric property of the camera. The focal lengths f_x and f_y correspond to the distance from the optical center, or pinhole (origin of the camera reference frame) to the image plane. This distance is measured in pixel units. In a perfect pinhole camera model both values must be equal which results in square image pixels. However their values can differ due to undesirable distortions from the lens or errors in the calibration process. The principal point c_x and c_y correspond to the closest projection of the optical center (perpendicular line) in the image plane, also measured in pixel units. In a perfect model this point should intersect the image

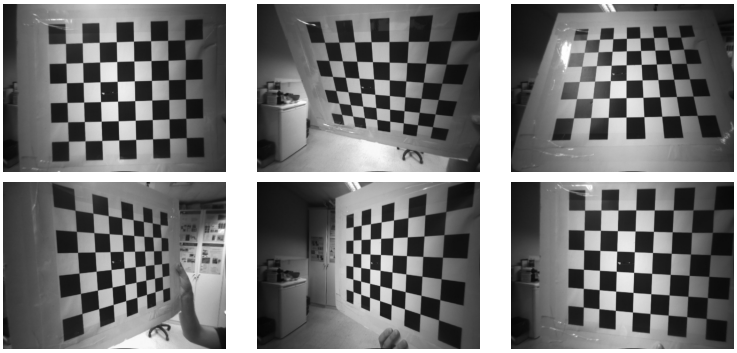


Figure 3.1: Calibration images, with a visible calibration pattern, used for the Bouguet Toolbox.

plane in the center of the image. Then the coordinates of these points are also affected by several factors that deviate it from its optimal location.

To map a 3D point with coordinates $P = [x, y, z] \in \mathbb{R}^3$ in metric units into 2D homogeneous image coordinates $p = [u, v] \in \mathbb{R}^2$ in pixel units, we apply the matrix K as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \tag{3.2}$$

Due to the use of lens in our cameras, the acquired images present undesired radial distortion, where straight lines appear as curves in the image. This type of distortion must be mapped so it can be corrected and the previous pinhole model can be applied. In this thesis we used [3] to completely calibrate the cameras by finding their intrinsic and radial distortion parameters. This library required several images of a chessboard acquired at different positions and orientations. After calibration, the radial distortion is corrected resulting in undistorted images, that will work as input for our calibration systems. The only parameters used in our systems are the intrinsic ones. Figure 3.1 shows an example of an image set used in our thesis to calibrate the intrinsic parameters of the cameras.

3.2 Epipolar Geometry

Assuming our cameras are approximated by the pinhole model, as described in (3.1), the epipolar geometry corresponds to the geometric relation between two images from

two cameras observing a scene from two different point of views. A point P is observed in one image (left) as p, as seen in figure 3.2.

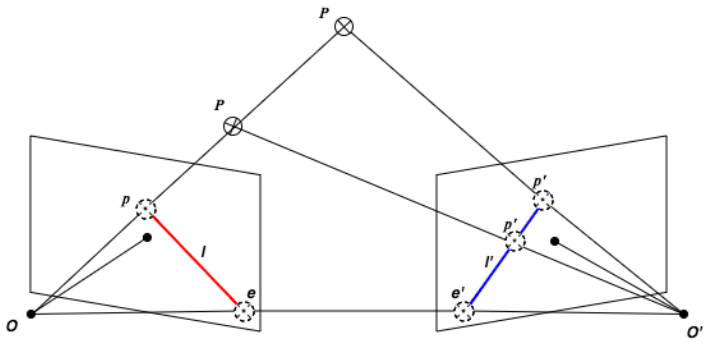


Figure 3.2: Epipolar geometry between two images.

However, all the points under the OP line will be projected to this image plane as p thus losing its depth in the operation. The only way to recover the point's depth is to observe the same scene from a different point of view, using a second camera (right). The projection of each point under OP will form a line l' in the right image, the blue line in figure 3.2. This line is called the epipolar line and it defines the set of possible locations where the point we acquired in the left image must lie on the right image. Every point from the left image has its corresponding epipolar line on the right image. The plane formed by the optical centers of both cameras, O and O', and the point P is called the epipolar plane which sets the constraint to the image points belonging to this plane, the epipolar constraint. The projection of each optical center in the other image is called the epipole, mapped in figure 3.2 as e and e'. It is possible that the epipole is placed outside the image plane, meaning one camera can not see the other. Nevertheless, every epipolar line pass through the epipole, regardless of its location.

In order to find the epipolar line for each point the roto-translation between the two cameras has to be known. This roto-translation will define a constraint between the matched points from the two images, the Fundamental Matrix Constraint. Let's assume the roto-translation between the left and right cameras is given by ${}^{R}T_{L}$, represented as:

$$ {}^{R}T_{L} = \begin{bmatrix} {}^{R}R_{L}\left(r_x, r_y, r_z\right) & {}^{R}t_{L} \\ \mathbf{0}^3 & 1 \end{bmatrix} \tag{3.3} $$

with ${}^{R}R_{L}\left(r_x, r_y, r_z\right)$ representing a rotation matrix and ${}^{R}t_{L} = [t_x, t_y, t_z]^T$. We can take the rotation matrix ${}^{R}R_{L}$, as well as the translational vector, ${}^{R}t_{L}$, and compute the

Essential Matrix E, [18], that relates corresponding points in both images:

$$E = \left[{}^{R}t_{L}\right]_{\times} {}^{R}R_{L} \tag{3.4}$$

where $\left[{}^{R}t_{L}\right]_{\times}$ is a skew-symmetric matrix using the vector components of ${}^{R}t_{L}$. From this, we can define the Essential Matrix Constraint, using homogeneous normalized coordinates m and m':

$$m'^{T} E m = 0 \tag{3.5}$$

where

$$m = K^{-1} p \tag{3.6}$$

with K corresponding to the intrinsic parameters of the camera (analogous for m'). To use homogeneous image coordinates of the points in pixel units, p and p', instead of their normalized coordinates, the Essential Matrix E must be converted to the Fundamental Matrix F, by applying the intrinsic parameters K of each camera as:

$$F = K'^{-T} E K^{-1} \tag{3.7}$$

By taking the image points p and p', the Fundamental Matrix Constraint is finally given by the following relation, which must hold for each pair of points if the roto-translation between the two cameras is correct:

$$p'^{T} F p = 0 \tag{3.8}$$

3.3 Stereo Rectification

During the stereo calibration procedure we need to compute the disparity from pairs of points extracted from non-parallel stereo images pairs. We use the stereo rectification algorithm described in [13] to bring the stereo images to a parallel rectified configuration. This algorithm takes the roto-translation between the two cameras and creates two projection matrices, one for each camera, that will force parallel epipolar lines by rotating them around their optical centers until both focal planes become coplanar.

Let's consider a generic stereo model where the rotation and translation from camera 1 to camera 2 are represented as ${}^{2}R_{1}$ and ${}^{2}t_{1}$, respectively. First we need to compute the rotation R_{rect} that will take both epipoles e_1 and e_2 (as seen in figure 3.2) to infinity and align the epipolar lines horizontally. We want the new x axis of the cameras to be

along the translation vector ${}^2t_1$, being parallel to the baseline, so the first row of R_{rect}, r_1 is given by:

$$r_1 = {}^2 t_1^T / \left\| {}^2t_1 \right\| \tag{3.9}$$

For the new y axis, we choose a vector that is orthogonal to both r_1 and to a vector k, which is an arbitrary unit vector that fixes the position of the new y axis in the plane orthogonal to x (usually we take k as a unit vector representing the old z axis of camera 1). The second row of R_{rect}, r_2, is then given by:

$$r_2 = k \times r_1 \tag{3.10}$$

The third row of R_{rect}, r_3, is the vector orthogonal to both r_1 and r_2, $r_3 = r_1 \times r_2$, thus completing the rows of R_{rect}:

$$R_{rect} = \left[r_1^T, r_2^T, r_3^T \right]^T \tag{3.11}$$

After applying R_{rect} to both cameras, we apply ${}^2R_1$ to camera 2 only to complete the alignment of the epipolar lines. The complete rectification matrices for each camera are given by:

$$\begin{aligned} R_1 &= R_{rect} \\ R_2 &= R_{rect} \left({}^2R_1 \right) \end{aligned} \tag{3.12}$$

(in some implementations, such as [3], the rotation ${}^2R_1$ is split in half and applied separately to both cameras to minimize the distortion).

A generic image point p_1, seen from camera 1, is rectified by first applying the rotation R_1, $q_1 = R_1 \left(K_1\right)^{-1} p_1$, where K_1 corresponds to its camera matrix. The point q_1 is then reprojected into the image plane, q_1' (divide each coordinate by the $q_1(3)$ coordinate) and its image coordinates p_1' are recomputed, $p_1' = K_1 q_1'$ (this process is analogous for camera 2). The vertical coordinates v of each p_1' and p_2', should be equal after the rectification.

After rectification the stereo points have perfectly horizontal epipolar lines with vertical component identical to the corresponding point's vertical coordinate. The depth Z of a point P_1 (seen from camera 1) can be directly obtained from triangulation [17], as a relation between the camera's baseline B, the new horizontal focal length f_x and

the horizontal disparity from rectified horizontal coordinates, $d_h = u_1 - u_2$:

$$Z = Bf_x/d_h \tag{3.13}$$

3.4 Stereo Triangulation

To reconstruct 3D points from non-parallel stereo images pairs (without performing stereo rectification as described in the previous section), we used the linear triangulation method described in [17]. This method provides an estimate that may not be optimal if i) the matched points are noisy which results in errors in the reconstructed points or; ii) the stereo roto-translation is noisy, which results in errors in the epipolar geometry. If none of these cases occur, the reconstructed point will be correct since both optical rays intersect in space at the precise location of the point.

Let's consider two matched image measurements p and p' acquired from a stereo pair, whose roto-translation is given by a rotation matrix R and a translation vector t. Each image measurement can be obtained from $p = QP$ and $p' = Q'P$ which corresponds to a projection of the same point P in the two images using the corresponding camera matrices Q and Q', that are given by:

$$\begin{aligned} Q &= K\,[I|0] \in \mathbb{R}^{3\times 4} \\ Q' &= K'\,[R|t] \in \mathbb{R}^{3\times 4} \end{aligned} \tag{3.14}$$

To get rid of the homogeneous scale factor we perform a cross product between p and QP, $p \times QP$ (analogous for p' and $Q'P$) which provides 3 equations for each image point:

$$\begin{aligned} p_1\,(Q_3P) - (Q_1P) &= 0 \\ p_2\,(Q_3P) - (Q_2P) &= 0 \\ p_1\,(Q_2P) - p_2\,(Q_1P) &= 0 \end{aligned} \tag{3.15}$$

where Q_i corresponds to the ith row of Q.

These equations can be combined into a form $AP = 0$ which is linear in P. Only the first two equations are linearly independent so we can discard the third one. Thus, considering both measurements p and p', our matrix A is given by:

$$A = \begin{bmatrix} p_1 Q_3 - Q_1 \\ p_2 Q_3 - Q_2 \\ p'_1 Q'_3 - Q'_1 \\ p'_2 Q'_3 - Q'_2 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{3.16}$$

We can now solve the equation $AP = 0$ for P by using the DLT (Direct Linear Transformation) method where we apply an SVD (Singular Value Decomposition) on matrix A. The SVD will decompose A in three matrices, $A = U\Sigma V^T$, with U and V^T corresponding to unitary matrices and the eigenvalues of A represented in matrix Σ. By taking the last column of V, corresponding to the unit singular vector associated to the smallest singular value, we obtain the solution for P:

$$P = V^4 / V_4^4 \tag{3.17}$$

where V^j corresponds to column j of V and V_i^j corresponds to a point of V at column j and row i.

3.5 Image Features

The visual information used in our calibration systems consists of image features directly obtained from the acquired images. In case of the head calibration system we are tracking image features between two consecutive time instants for the same camera. Considering that images are acquired at 30fps, the feature displacement in the images is small which allows the use of a simple feature detector (corners) since matching can be done by nearest neighbour. For the stereo calibration problem we must correctly match features between the two images where there is relevant displacement and where the stereo transformation may generate large perspective differences thus requiring more robust features, like SIFT features [26].

3.5.1 Harris Corner Detector

The Harris Corner Detector, as the name suggests, detects corners on the images. This method was developed by [16] and addressed in many other works, such as [39]. It uses intensity levels to detect corner regions that obey to the proposed model. The initial motivation was to obtain "good features to track" reason why they employed correlation in their analysis, where corner points stood-out.

Let's consider a grayscale image I. If we take an image patch over the area (u, v) and shift it by (x, y), the weighted sum of square differences, S, between the two image patches is given by:

$$S(x,y) = \sum_u \sum_v w(u,v) \left(I(u+x, v+y) - I(u,v)\right)^2 \tag{3.18}$$

where $w(u,v)$ corresponds to a weight window centered at (u,v), usually rectangular or gaussian. This function can be approximated by a Taylor expansion which results in the following expression:

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} M \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.19}$$

where M is a Hessian matrix, given by:

$$M = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3.20}$$

with I_x and I_y corresponding to the image derivatives in x and y directions respectively. This matrix corresponds to the Harris matrix and from its analysis it is possible to determine if the window in question contains a corner or not. By analysing the eigenvalues of M it is possible to distinguish between corers, edges and flat regions. Since this implies the computation of the eigenvalues which was computationally expensive by the time the algorithm was proposed, the authors came up with an alternative score to detect points of interest using the trace and determinant of M instead. If λ_1 and λ_2 are the eigenvalues of M, we have $det(M) = \lambda_1 \lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$. The corner score is given by:

$$R = det(M) - k\left(trace(M)\right)^2 \tag{3.21}$$

where k corresponds to a sensitivity constant that usually varies between 0.04 and 0.06 (the lowest the value the more sensitive the system in detecting corners).

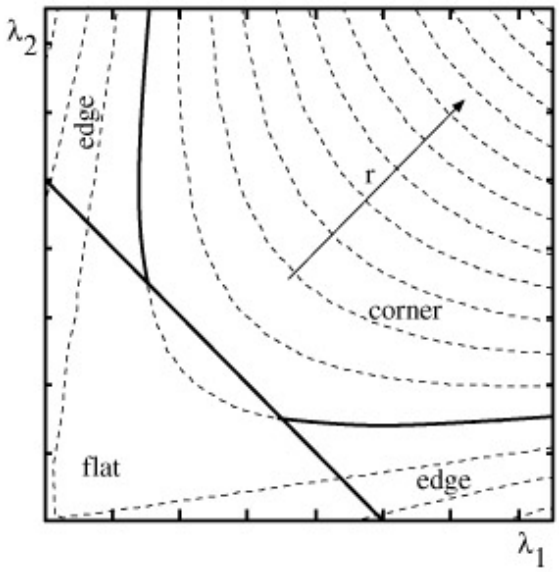


Figure 3.3: Region classification based on the score of R considering the eigenvalues of M, λ_1 and λ_2 ([11])

The values of the score R define whether a certain region is a corner, an edge or a flat surface, as seen in figure 3.3: if $|R|$ is small, the region is flat; if $R < 0$ the region is an edge; if R is large the region is a corner. The end result of this algorithm is an image with these scores where a threshold must be applied to find the points with the highest scores corresponding to good points of interest (image corners).

3.5.2 Normalized Cross Correlation

When we use the Harris Corner Detector, the feature descriptor corresponds to an image patch of size p extracted from the image and centered at the corner point (x, y). To compare two image patches and see if there is a match we use the Normalized Cross Correlation (NCC) algorithm, [5], which gives a score between 0 and 1 of how close two image patches are. Considering we have two image patches f and g, with the same size, the NCC is given by:

$$NCC = \frac{1}{n}\sum_{x,y}\frac{\left(f\left(x,y\right)-\bar{f}\right)\left(g\left(x,y\right)-\bar{g}\right)}{\sigma_f\sigma_g} \tag{3.22}$$

where $\bar{f}$ and $\bar{g}$ correspond to the average of f and g respectively, n is the total number of pixels in each patch and σ_f and σ_g are the standard deviations of f and g respectively, obtained using its standard formula as the square root of the variance of the

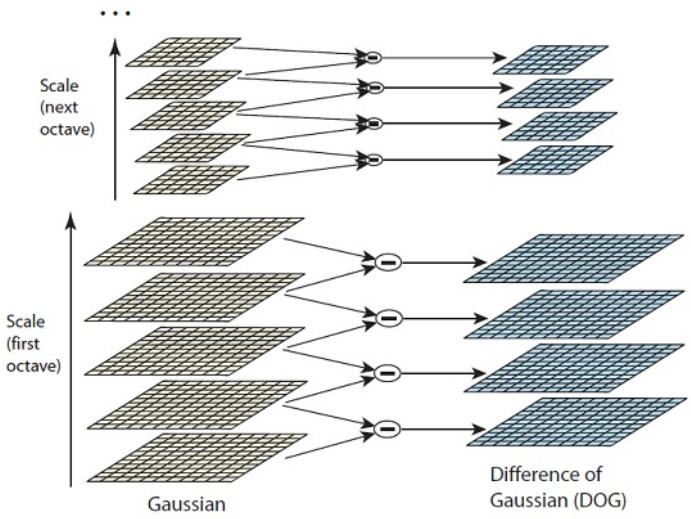


Figure 3.4: Representation of Difference of Gaussians (DoG) for each octave of scale space, after convolving the initial image with Gaussians ([26])

corresponding function.

3.5.3 SIFT

The SIFT (Scale-Invariant Features Transform), are extremely robust images features presented in [26]. These features are robust to scaling and rotation thus preserving their characteristics under scaled-euclidean transformations, which stands from all the other existing image features. The algorithm can be separated in two main steps: keypoint detection and descriptor extraction.

The SIFT algorithm, like the Harris Corner Detector, detects corners on the image. However, unlike the first, this method searches for corners under different scales of the image, using scale-space filtering. The original algorithm proposes to compute the Laplacian of Gaussians (LoG) for the image with various values of σ, which acts as a scale parameter. A gaussian kernel with low σ can detect small corners on the image while a high σ is suitable to detect large corners. This way it is possible to find local maxima for each scale σ and space (x, y) which correspond to a keypoint for that particular scale and space. However, most SIFT implementations use the Difference of Gaussians (DoG) as an approximation to the LoG since the last one requires higher computational power. The DoG is obtained for different octaves of the image in Gaussian Pyramid. The algorithm then detects local extrema for the different scales, finding candidate keypoints to be further verified, as demonstrated in figure 3.4.

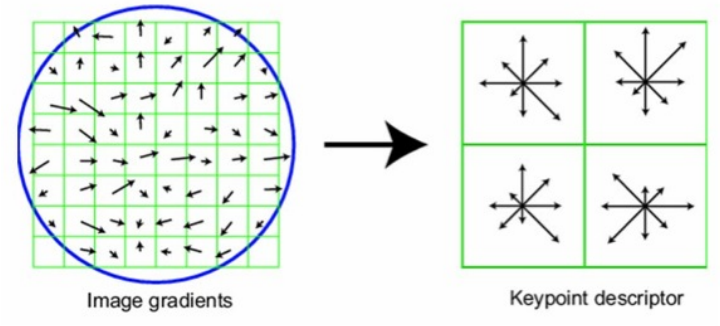


Figure 3.5: Computing the SIFT keypoint descriptor ([26])

Different thresholds are used to eliminate weak local extrema based on their intensity values (contrast threshold) and to get rid of edges that may appear, using an Hessian matrix similar to the one described in (3.20). The remaining points are strong candidates to become the final keypoints. In order to achieve invariance to rotation, the orientation of each keypoint is obtained by computing the gradient direction and magnitude under a certain region around each point, as seen in figure 3.5. The result is an orientation histogram with 36 bins, covering the 360deg, where the highest peak is selected to determine the orientation of that particular keypoint.

Each keypoint is represented by a 128 bin descriptor. This descriptor is obtained by creating a 16×16pixel neighbourhood around the keypoint, sub-divided into 16 blocks of 4×4pixel. For each block an 8 bin orientation histogram is created which results in the 128 bin values descriptor, as represented in figure 3.5. To match keypoints we find their k-nearest neighbours (with $k = 2$), by comparing their Hamming distances. In some cases the second closest neighbour is very close to the first one and only when the ratio between the two distances is larger than a certain threshold a match is assigned to the keypoints pair.

3.6 IEKF - Implicit Extended Kalman Filter

The Implicit Extended Kalman Filter (IEKF) is a variation of the Extended Kalman Filter (EKF) where the measurements take the form of an implicit constraint that is a function of both the system state and the sensors observations. Soatto introduced this variation in [40, 41] as a way to estimate the motion of a moving camera using the epipolar constraint, described in (3.8), as the system's measurements. The IEKF, as the EKF, makes some assumptions in the system model and the observations under which

the estimates converge. However, it is not guaranteed that some or all of these assumptions are fully satisfied in practice. For instance, the IEKF assumes the process noise and the measurement noise are both additive and uncorrelated zero-mean Gaussian process noises. Yet, in [40, 41] and in other works using the same implementation, such as [45], the observations are taken between two consecutive time instances which makes the measurement noises strongly correlated within one measurement time step, as stated in [45]. Although we didn't explore this solution in our thesis, it is possible to decorrelate these two noises, as showed in [41]. The gain in the system's performance didn't compensate the increase in the filter's complexity by applying the proposed solution which favored our decision in case of the head calibration system where the used measurements are taken in different time instants. In case of the stereo calibration system this assumption is not violated since all the measurements are taken within the same time instant and their noise is totally uncorrelated from the process noise. Another assumption that may not be satisfied corresponds to the implicit measurements noise which is approximated by an additive zero-mean Gaussian process. Even though the real observations noise may satisfy this assumption, it is not guaranteed that the implicit measurements noise will still obey to the noise model. In the following section we will explain in detail the whole implementation of the IEKF and address each one of these points.

3.6.1 State and Observation Model

As in the EKF, a generic IEKF without input considers the following transition model:

$$x^{k+1} = f^k\left(x^k\right) + w^k \tag{3.23}$$

where f^k corresponds to the state transition function, x^k and x^{k+1} denote the system states at time instants k and $k+1$ and $w^k \sim \mathcal{N}\left(0, Q^k\right)$ where Q^k represents the covariance matrix of the zero mean state transition noise w^k, assumed to be an additive zero-mean Gaussian process.

In the standard EKF the measurements are explicit functions of the system state x and they can be obtained from a measurement model of the form:

$$y^{k+1} = h^{k+1}\left(x^{k+1}\right) + v^{k+1} \tag{3.24}$$

where h^{k+1} corresponds to the measurement function, with $v^{k+1} \sim \mathcal{N}\left(0, R^{k+1}\right)$ where R^{k+1} represents the covariance matrix of the measurement noise v^k, assumed to be an additive zero-mean Gaussian process. However in the IEKF, the filter measurement

equation takes the form of a constraint z that must be fulfilled and is an implicit function of both the system state x and the physical measurements from the sensors, y:

$$z^{k+1} = h^{k+1}\left(x^{k+1}, y^{k+1}\right) = 0 \tag{3.25}$$

This hybrid model is extremely usefull in cases where the system's estimation error, represented in z^{k+1}, cannot be obtained from a simple subtraction between a function of the system state $h^{k+1}\left(x^{k+1}\right)$, and an observation y^{k+1}, like $z^{k+1} = y^{k+1} - h^{k+1}\left(x^{k+1}\right)$.

3.6.2 Prediction and Update Equations

The IEKF, like the EKF, is a two-step procedure with a prediction and an update step. In the prediction step, the system state is propagated using the dynamic model described in (3.23):

$$\bar{x}^{k+1} = f^k\left(\hat{x}^k\right) \tag{3.26}$$

where $\hat{x}^k$ corresponds to the estimate of x from the previous time instant, with the previous state covariance matrix estimation $\hat{P}^{k+1}$ being propagated and predicted using the standard filter equation:

$$\bar{P}^{k+1} = F^k \hat{P}^k \left(F^k\right)^T + Q^k \tag{3.27}$$

where F^k is the Jacobian of f^k obtained by linearizing the function, evaluated at the previous state estimate $\hat{x}^k$:

$$F^k = \left.\frac{\partial f^k}{\partial x}\right|_{x=\hat{x}^k} \tag{3.28}$$

In the update step, the predicted filter measurement $\bar{z}^{k+1}$ is obtained from the current state prediction $\bar{x}^{k+1}$ and the sensors measurements y^{k+1} :

$$\bar{z}^{k+1} = h^{k+1}\left(\bar{x}^{k+1}, y^{k+1}\right) \tag{3.29}$$

The final state update $\hat{x}^{k+1}$, given the previous measurement prediction, takes the form:

$$\begin{aligned} \hat{x}^{k+1} &= \bar{x}^{k+1} + K^{k+1}\left(z^{k+1} - \bar{z}^{k+1}\right) \\ &= \bar{x}^{k+1} - K^{k+1}\bar{z}^{k+1} \end{aligned} \tag{3.30}$$

where the matrix K^{k+1} corresponds to the Kalman gain (note that as described in (3.25), $z^{k+1} = 0$). The update of the state covariance matrix $\hat{P}^{k+1}$ is given by:

$$\begin{aligned} \hat{P}^{k+1} &= \left(I - K^{k+1}H^{k+1}\right)\bar{P}^{k+1}\left(I - K^{k+1}H^{k+1}\right)^T \\ &\quad + K^{k+1}\tilde{R}^{k+1}\left(K^{k+1}\right)^T \end{aligned} \tag{3.31}$$

where the Kalman gain matrix K^{k+1} is obtained as:

$$K^{k+1} = \bar{P}^{k+1}H^{k+1}\left(H^{k+1}\bar{P}^{k+1}\left(H^{k+1}\right)^T + \tilde{R}^{k+1}\right)^{-1} \tag{3.32}$$

In the previous equations, H^{k+1} corresponds to the Jacobian of h^{k+1} obtained through linearization of the function, evaluated at the current state estimate $\hat{x}^{k+1}$:

$$H^{k+1} = \left.\frac{\partial h^{k+1}}{\partial x}\right|_{x=\hat{x}^{k+1}} \tag{3.33}$$

and $\tilde{R}^{k+1}$ is the first-order approximation of the covariance of the noise in the implicit measurement constraint. The relation between this covariance noise and the covariance noise of the sensors measurements R^{k+1} is given by:

$$\tilde{R}^{k+1} = D^{k+1}R^{k+1}\left(D^{k+1}\right)^T \tag{3.34}$$

where D^{k+1} is given by:

$$D^{k+1} = \left.\frac{\partial h^{k+1}}{\partial y}\right|_{y=y^{k+1}} \tag{3.35}$$

The pseudocode simulating one step of a generic IEKF filter is presented next.

Algorithm 1 IEKF generic implementation (pseudo-code)

Initialize x^0, P^0, Q^0 and R^0
while $i < N$ **do**

Prediction Step

$\bar{x}^{i+1} = f\left(\hat{x}^i\right)$
$F^i = Jacobian\left(f, \hat{x}^i\right)$
$\bar{P}^{i+1} = F^i \hat{P}^i \left(F^i\right)^T + Q^0$

Update Step

$y^{i+1} = Observations\,()$
$\bar{z}^{i+1} = h\left(\bar{x}^{i+1}, y^{i+1}\right)$
$H^{i+1} = Jacobian\left(h, \bar{x}^{i+1}\right)$
$D^{i+1} = Jacobian\left(h, y^{i+1}\right)$
$\tilde{R}^{i+1} = D^{i+1} R^0 \left(D^{i+1}\right)^T$
$K^{i+1} = \bar{P}^{i+1} H^{i+1} \left(H^{i+1} \bar{P}^{i+1} \left(H^{i+1}\right)^T + \tilde{R}^{i+1}\right)^{-1}$
$\hat{x}^{i+1} = \bar{x}^{i+1} - K^{i+1} \bar{z}^{i+1}$
$\hat{P}^{i+1} = \left(I - K^{i+1} H^{i+1}\right) \bar{P}^{i+1} \left(I - K^{i+1} H^{i+1}\right)^T + K^{i+1} \tilde{R}^{i+1} \left(K^{i+1}\right)^T$

$i = i + 1$

end while

3.7 Finite-Difference Approximations of Derivatives

A finite difference method is a numerical method for solving differential equations. In our thesis we used this method to obtain the Jacobians corresponding to the first-order derivatives of the functions f^k, in 3.26 and h^{k+1}, in 3.29, evaluated at the corresponding points.

Let's consider a generic function $f : \mathbb{R}^N \to \mathbb{R}^M$. The jacobian F of such a function evaluated at a point x_0 is given by:

$$F = \left.\frac{\delta f}{\delta x}\right|_{x=x_0} = \left.\begin{bmatrix} \frac{\delta f_1}{\delta x_1} & \cdots & \frac{\delta f_1}{\delta x_N} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_M}{\delta x_1} & \cdots & \frac{\delta f_M}{\delta x_N} \end{bmatrix}\right|_{x=x_0} \tag{3.36}$$

where each partial derivative is obtained using approximation given by the finite differ-

ence method for a small ε:

$$F_{ij} = \left.\frac{\delta f_i}{\delta x_j}\right|_{x_j=a} \approx \frac{f_i\left(a+\varepsilon\right) - f_i\left(a-\varepsilon\right)}{2\varepsilon} \tag{3.37}$$

This approximation is very useful when the function is extremely complex, such as the ones used in our thesis where the analytical expression is hard to obtain. In such cases it is common to approximate the real derivative using this method with the disadvantage of increasing the processing time (although in some particular cases where the differential function takes more time to evaluate than the original one).

3.8 Rodrigues Rotation Formula

The Rodrigues Rotation Formula, [37] as cited in [7], is an algorithm for computing a rotation matrix in $SO\left(3\right)$ corresponding to a rotation about a vector in space given an axis ω and angle of rotation θ. From the three rotation values around each axis, r_x, r_y and r_z, the rotation matrix R can be obtained from the following equation:

$$R = I + \left[\Omega\right]_\times \sin\left(\theta\right) + \left[\Omega\right]_\times^2 \left(1 - \cos\left(\theta\right)\right) \tag{3.38}$$

where $\theta = \sqrt{r_x^2 + r_y^2 + r_z^2}$ and $\left[\Omega\right]_\times$ corresponds to a skew-symmetric matrix constructed from a normalized rotation vector ω whose components are given by $\omega_x = r_x/\theta$, $\omega_y = r_y/\theta$ and $\omega_z = r_z/\theta$. The components of the rotation matrix R are explicitly given by:

$$\begin{aligned}
R_{11} &= \cos\theta + \omega_x^2\left(1 - \cos\theta\right) \\
R_{12} &= \omega_x\omega_y\left(1 - \cos\theta\right) - \omega_z\sin\theta \\
R_{13} &= \omega_y\sin\theta + \omega_x\omega_z\left(1 - \cos\theta\right) \\
R_{21} &= \omega_z\sin\theta + \omega_x\omega_y\left(1 - \cos\theta\right) \\
R_{22} &= \cos\theta + \omega_y^2\left(1 - \cos\theta\right) \\
R_{23} &= \omega_y\omega_z\left(1 - \cos\theta\right) - \omega_x\sin\theta \\
R_{31} &= \omega_x\omega_z\left(1 - \cos\theta\right) - \omega_y\sin\theta \\
R_{32} &= \omega_x\sin\theta + \omega_y\omega_z\left(1 - \cos\theta\right) \\
R_{33} &= \cos\theta + \omega_z^2\left(1 - \cos\theta\right)
\end{aligned} \tag{3.39}$$

with R_{ij} corresponding to the element at the ith row and jth column of R.

In the inverse operation, to obtain the initial rotation values r_x, r_y and r_z from a rotation matrix R, we calculate the angle of rotation of the trace of R:

$$\theta = \arccos\left(trace\left(R\right)\right) \tag{3.40}$$

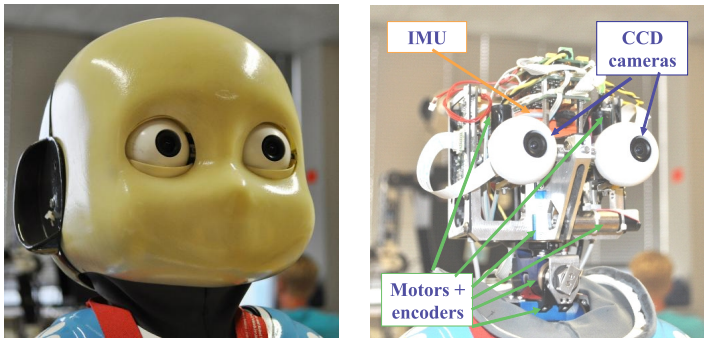
and use it to find the components of the normalized rotation vector ω:

$$\omega = \frac{1}{2\sin\left(\theta\right)} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \tag{3.41}$$

Chapter 4

Head Calibration System

In this chapter we will explain in detail the design and implementation of the real time head calibration system whose base is an Implicit Extended Kalman Filter, explained in section 3.6. This system assumes the robotic platform under calibration is equipped with three types of sensors: an inertial measurement unit (IMU) that generates linear acceleration and angular velocities measurements, motor encoders that provide the motor angles of the joints and stereo cameras that generate real images of the world, as seen in the following figure.



a) iCub (Chica) head b) iCub head sensors

Figure 4.1: The iCub robotic head with its embedded sensors.

If, by any chance, one of these sensors stops working, the system is still able to calibrate the robotic head using the other sensors up to the kinematic location of the failed sensors.

In the following implementation we will assume the base of the kinematic chain is static and aligned with gravity in a planar horizontal surface, the world is static and infinite (all the objects seen by the cameras are at a very large distance) and there are no

mounting errors of the IMU nor the cameras (the calibration of the cameras mounting errors will be addressed in Chapter 6).

4.1 State Transition Model

The system state x_H is given by:

$$x_H = \begin{bmatrix} \delta_0 & \dots & \delta_{N-1} \end{bmatrix} \in \mathbb{R}^N \tag{4.1}$$

where δ_i corresponds to the ith joint offset, represented in figure 2.1. These offsets are assumed to be almost constant over time, thus the state transition equation f_H corresponds to the identity. To allow for small changes of the values over time, e.g. due to mechanical wear or slippage, we allow for some state transition noise w^k.

The system state transition equation is therefore:

$$x_H^{k+1} = x_H^k + w^k \tag{4.2}$$

Here $w^k \sim \mathcal{N}\left(0, Q^k\right)$ where Q^k represents the covariance matrix of the zero mean state transition noise w^k. The system can be adapted to be more or less sensitive to variations in the estimate of the offsets by changing this covariance matrix.

4.1.1 System Initialization

Considering we are estimating the offsets of a N joints kinematic chain, the system state x_H^0 is initialized with the values of the encoders at start-up, y_E^0:

$$x_H^0 = y_E^0 \in \mathbb{R}^N \tag{4.3}$$

The covariance matrices P_H^0 and Q_H^0, corresponding to the system state uncertainty and the system uncertainty during the state transition process respectively, are both diagonal considering the joints offsets are independent. They are initialized with the standard deviation values set for the system state uncertainty σ_{Px}^0, or our confidence on the initial system state values, and for the system process noise σ_{Qx}^0, or how we believe the system state variables will change between two consecutive time instants:

$$\begin{cases} P_H^0 = \boldsymbol{I}^N . \left(\sigma_{Px}^0\right)^2 \in \mathbb{R}^{N \times N} \\ Q_H^0 = \boldsymbol{I}^N . \left(\sigma_{Qx}^0\right)^2 \in \mathbb{R}^{N \times N} \end{cases} \tag{4.4}$$

where $\boldsymbol{I}^N$ corresponds to an N size identity matrix and where we assume each parameter under estimation has the same level of uncertainty.

4.2 Observation Model

The considered robotic platforms are equiped with three types of sensors: an IMU, cameras and encoders. From these we can obtain four types of different measurements, depending on the sensor we are considering.

The IMU provides measurements for linear accelerations

$$y_A^{k+1} = \begin{bmatrix} a_x^{k+1} & a_y^{k+1} & a_z^{k+1} \end{bmatrix}^T \in \mathbb{R}^3$$

and angular velocities

$$y_W^{k+1} = \begin{bmatrix} w_x^{k+1} & w_y^{k+1} & w_z^{k+1} \end{bmatrix}^T \in \mathbb{R}^3$$

for the three principal axes on which it is mounted. We assume that the linear accelerations measured by the IMU correspond to the effects of the gravity vector decomposed in the three components of x, y and z affected by sensor noise, which is valid for slow movements of the robotic head.

The cameras provide M image features represented by their image coordinates

$$f_i = [u_i\,,\, v_i] \in \mathbb{R}^2$$

in pixel units. Here we are interested in image movement induced by the joint movement, hence we always consider a pair of consecutive frames as measurements

$$y_F^k = \begin{bmatrix} f_0^k & \dots & f_M^k \end{bmatrix}^T \in \mathbb{R}^{2M}$$

and

$$y_F^{k+1} = \begin{bmatrix} f_0^{k+1} & \dots & f_M^{k+1} \end{bmatrix}^T \in \mathbb{R}^{2M}$$

The encoders provide N measurements of the relative position of the joints in two consecutive time instants k and $k+1$,

$$y_E^k = \begin{bmatrix} e_0^k & \dots & e_N^k \end{bmatrix}^T \in \mathbb{R}^N$$

and

$$y_E^{k+1} = \begin{bmatrix} e_0^{k+1} & \dots & e_N^{k+1} \end{bmatrix}^T \in \mathbb{R}^N$$

where e_i corresponds to the ith relative encoder measurement as represented in figure 2.1b). The encoders usually work at a very high frequency (1000Hz), much higher than the IMU (100Hz) or the cameras (30Hz), the slower sensor. For this reason, the IMU and cameras are responsible for setting the sampling instants where the measurements are acquired since there are always available encoders measurements. Considering the IMU works at a higher frequency than the cameras, its measurements are acquired in between two camera measurements.

In order to predict all the sensors measurements an estimate of the absolute value of each joint is needed to compute the complete transformation ${}^iT_0$ from the base of the kinematic chain to the reference frame i where each sensor is mounted. Because we assume that points are distant and generate low parallax, only the rotational part ${}^iR_0$ will be used. Collecting equations (2.1) in vector form, the absolute values of the joints Θ_H for both time instants k and $k+1$ are given by

$$\Theta_H^{k+1} = y_E^{k+1} - \bar{x}_H^{k+1} \tag{4.5}$$

where the current offsets prediction $\bar{x}_H^{k+1}$ was used in the two cases because it is the most up to date value available of parameter that, ideally, should be constant over time.

Considering the IMU is mounted on reference frame I we represent the base to IMU coordinate transformation by ${}^IR_0\left(\Theta_H^{k+1}\right)$. The prediction of the linear accelerations measurements $\bar{y}_A^{k+1}$ are obtained by mapping the world constant gravity vector by this rotation:

$$\bar{y}_A^{k+1} = {}^IR_0\left(\Theta_H^{k+1}\right) . \begin{bmatrix} 0 & -G & 0 \end{bmatrix}^T \tag{4.6}$$

where G corresponds to the standard gravity value $9.806 m.s^{-2}$.

The predictions of the angular velocity measurements $\bar{y}_W^{k+1}$ are computed from the derivative of the IMU reference frame, here approximated by the change of this reference frame between two consecutive time instants divided by the change in time. Since the base of the robotic platform does not move, these velocities can be obtained by

$$\bar{y}_W^{k+1} = Rodrigues^{-1}\left({}^IR_0\left(\Theta_H^{k+1}\right) . \left({}^IR_0\left(\Theta_H^{k}\right)\right)^{-1}\right) / dT \tag{4.7}$$

where the inverse Rodrigues function described in section 3.8 provides the instant angular

change and dT is the time interval between the two encoder measurements.

To obtain the image features predictions $\bar{y}_F^{k+1}$ for a camera in reference frame C we need as well two consecutive orientations of this reference frame. Since we consider low parallax, the image features will only be affected by rotation. The prediction of the image features (in normalized metric coordinates) is obtained as:

$$\bar{y}_F^{k+1} = \begin{bmatrix} \bar{f}_0^{k+1} & \dots & \bar{f}_{M-1}^{k+1} \end{bmatrix}^T \tag{4.8}$$

where the ith image feature prediction $\bar{f}_i^{k+1}$ is given by

$$\begin{bmatrix} \lambda \bar{f}_i^{k+1} & \lambda \end{bmatrix}^T = {}^C R_0 \left(\Theta_H^{k+1} \right) . \left({}^C R_0 \left(\Theta_H^k \right) \right)^{-1} . \begin{bmatrix} f_i^k & 1 \end{bmatrix}^T \tag{4.9}$$

where λ is a scale factor. This equation provides a set of constraints, two for each image feature, that needs to be satisfied by the measurements y_F^{k+1} in the Implicit Kalman Filter Formulation.

Since the IMU seldomly works at the same frequency as the image acquisition sensors, these readings are usually not simultaneously available. Hence at each time step we either have an IMU observation or an image observation which needs to be filtered. The system measurements constraints $\bar{z}_H^{k+1}$, as described in equation (3.29) are thus at each time step k given by one of two possibilities:

$$\bar{z}_H^{k+1} = \begin{cases} h_H^{k+1} \left(\bar{x}_H^{k+1}, y_A^{k+1}, y_W^{k+1}, y_E^k, y_E^{k+1} \right) = \begin{bmatrix} \bar{z}_A^{k+1} & \bar{z}_W^{k+1} \end{bmatrix} + \tilde{v}_I^{k+1} & \text{if IMU sample} \\ \\ h_H^{k+1} \left(\bar{x}_H^{k+1}, y_F^k, y_F^{k+1}, y_E^k, y_E^{k+1} \right) = \begin{bmatrix} \bar{z}_F^{k+1} \end{bmatrix} + \tilde{v}_C^{k+1} & \text{if vision sample} \end{cases} \tag{4.10}$$

with

$$\begin{aligned} \bar{z_A}^{k+1} &= y_A^{k+1} - \bar{y}_A^{k+1} \\ \bar{z_W}^{k+1} &= y_W^{k+1} - \bar{y}_W^{k+1} \\ \bar{z_F}^{k+1} &= y_F^{k+1} - \bar{y}_F^{k+1} \end{aligned} \tag{4.11}$$

and where $\tilde{v}_I^{k+1} \sim \mathcal{N}\left(0, \tilde{R}_I^{k+1}\right)$ and $\tilde{v}_C^{k+1} \sim \mathcal{N}\left(0, \tilde{R}_C^{k+1}\right)$ are the observation noises of the implicit measurement constraint in case of IMU or image measurements respectively, assumed to be a zero mean Gaussian with covariance matrix $\tilde{R}_I^{k+1}$ or $\tilde{R}_C^{k+1}$, obtained using the equation described in (3.34). Depending on the observation samples, we must initialize the system's observation covariance matrix R^{k+1} differently, using the standard deviation values for the different observations noise obtained from a direct analysis of the

sensors, σ^0_{Ra} (linear acceleration), σ^0_{Rw} (angular velocities), σ^0_{Rf} (image features), σ^0_{Re} (encoders). In case we have an IMU observation sample, the covariance matrix R_I^{k+1} is given by:

$$R^{k+1} = R_I^{k+1} = \begin{bmatrix} R_A^{k+1} & 0 & \dots & 0 \\ 0 & R_W^{k+1} & 0 & \vdots \\ \vdots & 0 & R_E^{k+1} & 0 \\ 0 & \dots & 0 & R_E^{k+1} \end{bmatrix} \in \mathbb{R}^{(6+2N)\times(6+2N)} \tag{4.12}$$

where

$$\begin{cases} R_A^{k+1} = \boldsymbol{I}^3 \left(\sigma^0_{Ra}\right)^2 \in \mathbb{R}^{3\times3} \\ R_W^{k+1} = \boldsymbol{I}^3 \left(\sigma^0_{Rw}\right)^2 \in \mathbb{R}^{3\times3} \\ R_E^{k+1} = \boldsymbol{I}^N \left(\sigma^0_{Re}\right)^2 \in \mathbb{R}^{N\times N} \end{cases} \tag{4.13}$$

Even though the encoders are obtained in two consecutive time instants, their observation noise is assumed to be the same, reason why we repeat the covariance matrix R_E^{k+1} in R^{k+1}.

In the case where we have a vision observation sample, the covariance matrix R_C^{k+1} is given by:

$$R^{k+1} = R_C^{k+1} = \begin{bmatrix} R_F^{k+1} & 0 & 0 \\ 0 & R_E^{k+1} & 0 \\ 0 & 0 & R_E^{k+1} \end{bmatrix} \in \mathbb{R}^{(2M+2N)\times(2M+2N)} \tag{4.14}$$

where

$$\begin{cases} R_F^{k+1} = \boldsymbol{I}^{2M} \left(\sigma^0_{Rf}\right)^2 \in \mathbb{R}^{2M\times2M} \\ R_E^{k+1} = \boldsymbol{I}^N \left(\sigma^0_{Re}\right)^2 \in \mathbb{R}^{N\times N} \end{cases} \tag{4.15}$$

The final covariance matrix corresponding to the implicit measurements constraints, $\tilde{R}^{k+1}$, is obtained from equation (3.34) using the correct matrix R^{k+1} already described.

To better understand the head calibration system, we have created a pseudo-code simulating one step of the filtering process, from a total of I iterations for a generic kinematic chain with N joints. The pseudo-code is implementing the head calibration system using all the information described in this chapter jointly with the standard IEKF equations defined in section 3.6.

Algorithm 2 Head Calibration System (pseudo-code)

Define σ^0_{Qx}, σ^0_{Px}, σ_{Ra}, σ_{Rw}, σ_{Rf} and σ_{Re} ▷ Standard deviation values

Initialization

$x^0_H = zeros(1, N)$

$P^0_H = eye(N)(\sigma^0_{Px})^2$ ▷ Eq.4.4

$Q^0_H = eye(N)(\sigma^0_{Qx})^2$ ▷ Eq.4.4

while $i < I$ **do**

Prediction Step

$\bar{x}^{i+1}_H = f_H\left(\hat{x}^i_H\right)$ ▷ Eq.4.2

$F^i_H = Jacobian\left(f_H, \hat{x}^i_H\right)$ ▷ Eq.3.36

$\bar{P}^{i+1}_H = F^i_H \hat{P}^i_H \left(F^i_H\right)^T + Q^0_H$

Update Step

if got vision sample? **then**

$y^{i+1}_F = ImageFeatures\,()$

$\left[y^i_E, y^{i+1}_E\right] = Encoders\,()$

$y^{i+1} = \left(y^{i+1}_F, y^i_E, y^{i+1}_E\right)$

$R^{i+1}_F = eye(2M)(\sigma_{Rf})^2$

$R^{i+1}_E = eye(N)(\sigma_{Re})^2$

$R^{i+1} = diag(R^{i+1}_F, R^{i+1}_E, R^{i+1}_E)$

else ▷ got IMU sample

$y^{i+1}_A = LinearAcceleration\,()$

$y^{i+1}_W = AngularVelocities\,()$

$\left[y^i_E, y^{i+1}_E\right] = Encoders\,()$

$y^{i+1} = \left(y^{i+1}_A, y^{i+1}_W, y^i_E, y^{i+1}_E\right)$

$R^{i+1}_A = eye(3)(\sigma_{Ra})^2$

$R^{i+1}_W = eye(3)(\sigma_{Rw})^2$

$R^{i+1}_E = eye(N)(\sigma_{Re})^2$

$R^{i+1} = diag(R^{i+1}_A, R^{i+1}_W, R^{i+1}_E, R^{i+1}_E)$

end if

$\bar{z}^{i+1}_H = h_H\left(\bar{x}^{i+1}_H, y^{i+1}\right)$ ▷ Eq.4.10

$H^{i+1}_H = Jacobian\left(h_H, \bar{x}^{i+1}_H\right)$ ▷ Eq.3.36

$D^{i+1}_H = Jacobian\left(h_H, y^{i+1}\right)$ ▷ Eq.3.36

$\tilde{R}^{i+1} = D^{i+1}_H R^{i+1} \left(D^{i+1}_H\right)^T$

$K^{i+1} = \bar{P}^{i+1}_H H^{i+1}_H \left(H^{i+1}_H \bar{P}^{i+1}_H \left(H^{i+1}_H\right)^T + \tilde{R}^{i+1}\right)^{-1}$

$\hat{x}^{i+1}_H = \bar{x}^{i+1}_H - K^{i+1}\bar{z}^{i+1}_H$

$\hat{P}^{i+1}_H = \left(I - K^{i+1}H^{i+1}_H\right)\bar{P}^{i+1}_H \left(I - K^{i+1}H^{i+1}_H\right)^T + K^{i+1}\tilde{R}^{i+1}\left(K^{i+1}\right)^T$

$i = i + 1$

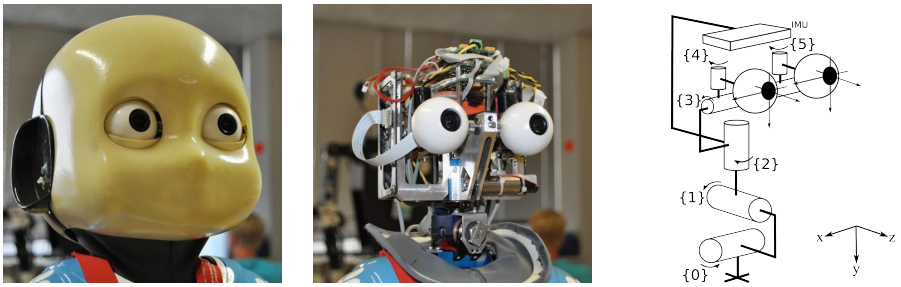
end while

Chapter 5

Head Calibration Results

In this chapter we will evaluate the proposed architecture for the head calibration system. We will perform simulated and real experiments to evaluate the system in terms of its accuracy and repeatability. The real experiments were performed using the iCub robotic platform, [27], more specifically the head [1]. The iCub was developed in the context of the EU project RobotCub and was adopted by more than 20 laboratories worldwide. The full robot has 53 DOF with the head having 5 DOF from the neck to each of the eyes, as seen in Fig. 5.1.

Each joint is equipped with relative encoders that are extremely precise but are unable to measure the absolute zero position of the joint. The head is also equipped with an Xsens IMU that measures the linear accelerations and angular velocities at a frequency of 100Hz. This sensor is placed on top of the head right before the eyes tilt joint, thus rigidly attached to frame {2}, as seen in Fig. 5.1c). Finally the head is equipped with two Pointgrey Dragonfly cameras, that work at 30Hz and provide RGB images with VGA resolution (640 × 480pixel). These cameras have 4.7 × 3.5mm CCD



a) Chica head b) Head structure c) Head kinematic model

Figure 5.1: The iCub robotic head used in our real head calibration experiments.

Parameter	Left Camera	Right Camera
Width (pixel)	640	640
Height (pixel)	480	480
f_x (pixel)	332.706	342.367
f_y (pixel)	382.658	389.545
c_x (pixel)	348.316	350.601
c_y (pixel)	241.872	251.704

Table 5.1: Intrinsic parameters of the cameras: resolution (Width and Height), focal lengths (f_x and f_y) and optical centers (c_x and c_y)

sensors and lenses that yield a field of view of 87.3° (horizontal) and 70.8° (vertical). The cameras are the last sensors in the kinematic chain being affected by all the joints. The intrinsic parameters of the cameras were calibrated using the Bouguet Toolbox, presented in section 3.1 and the radial distortion was compensated via unwrapping the radial image. The calibrated parameters are presented in the following table:

In order to validate the proposed architecture we performed simulated experiments where all the sensors measurements were simulated and fed to the system just like in the real case. For both cases, the simulated and real, we created several datasets. Each dataset contains information from all the sensors at every time step: the linear accelerations and angular velocities from the IMU, the motor encoders and stereo images from both cameras (or simulated image features in the simulation case).

The calibration procedure for the real case goes as follows: the motors are initialized at an arbitrary position and we start the data acquisition while randomly rotating the head, as seen in Fig. 5.2. An example of the calibration procedure can be seen in this video: `https://www.youtube.com/watch?v=mIn0sSke_kw`. Several datasets were acquired either for the same or for different encoder offsets, to completely evaluate the system in terms of its accuracy and repeatability.

To initialize the system state uncertainty as well as the transition process noise we took into consideration the nature of the problem. We are estimating joint offsets that must be combined with the encoders values to provide calibrated measurements. Since we are initializing the system state as a zero vector, as already mentioned in subsection 4.1.1, we have a large uncertainty at the beginning considering the physical limits of each joint. To ensure the system's convergence, the state's initial uncertainty must be large enough to include all the possible values the offsets could take. Therefore we initialized each uncertainty σ^0_{Px} with a large value, 40deg. The transition process noise σ^0_{Qx} was set to 0.05deg, as a trade off between fast enough convergence and low sensitivity to noise.

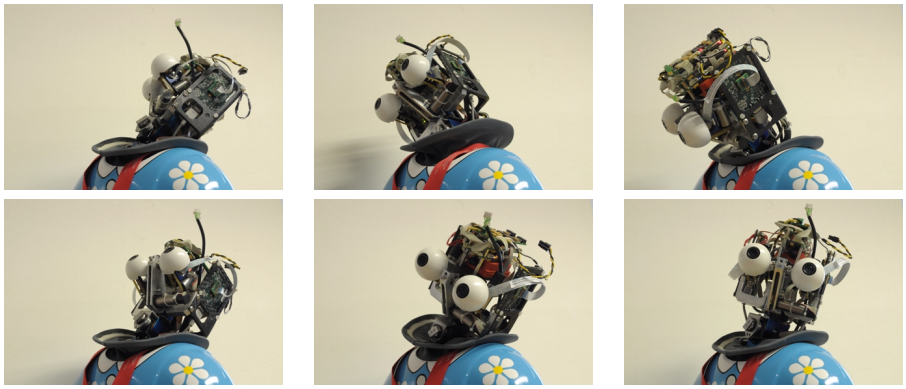


Figure 5.2: The head calibration procedure where the head is initialized at a random position and is rotated in order acquire data for the different experiments.

Sensor Measurement	Symbol	Noise Std. Dev	Frequency (Hz)
Linear Acceleration	σ^0_{Ra}	$0.0447 m/s^2$	100
Angular Velocities	σ^0_{Rw}	0.5deg/s	100
Encoders	σ^0_{Re}	0.028deg	> 100
Image Features	σ^0_{Rf}	2pixel	30

Table 5.2: Characterization of the Sensors

We analysed the noise levels that described each of the sensors' measurements, assumed to be zero mean Gaussian. The standard deviations values are represented in Table 5.2.

In case of the image features, for this system we used the Harris Corner Detector explained in subsection 3.5.1 and Normalized Cross Correlation to track features between two consecutive images. Fig. 5.3 shows an example of image features acquisition and corresponding tracking.

The features search in the next image was done within a limited region around the previous location of the features to reduce the computation time, assuming the images movement was small enough between two consecutive time instants.

5.1 Simulated Experiments

It is very difficult to measure the real absolute zero position of each motor joint considering there is no ground-truth for the real robot. One way to evaluate the calibration system in terms of its accuracy is by testing it with simulated experiments. Each experiment simulated the real conditions of the robot and introduced the same level of

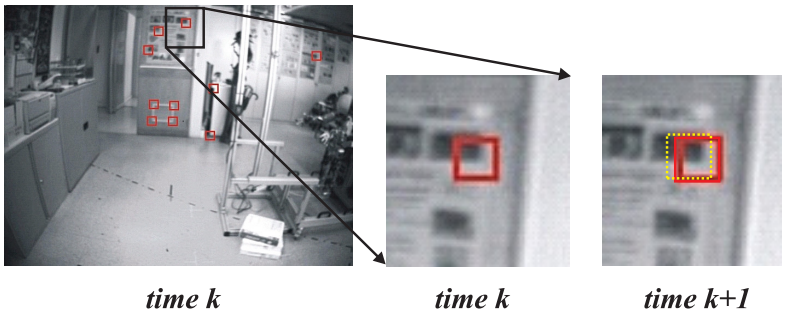


Figure 5.3: Example of image features acquisition and tracking using the Harris Corner Detector and Normalized Cross Correlation between two images obtained at consecutive time instants k and $k+1$.

noise for each sensor according to the values in Table 5.2. It is very important to create simulated environments whose conditions are similar to the real ones.

We performed 5 experiments where we initialized each joint with different offset values. For each experiment we performed 5 trials where we simulated the rotation of the robot head. Starting from an arbitrary position, the rotation step for an ith joint, λ_i^{step} was sampled from a Uniform Distribution $\lambda_i^{step} \sim \mathcal{U}(-\theta, \theta)$ with $\theta = 0.1$deg and a sample rate of 0.033s. These values were chosen in order to best replicate the real movement that was performed by hand. Between each two steps we sampled 50 virtual image features by first generating their $3D$ coordinates, within a virtual scenario ranging from 250mm to 4000mm, for a certain time instant and calculating their matched coordinates in the consecutive instant by using the complete roto-translation of the head. The virtual image features used as measurements were then obtained by projecting each $3D$ generated point into the corresponding virtual images. Zero-mean Gaussian noise was added to the matched image coordinates, sampled from normal distribution with a standard deviation of $\sigma_{Rf}^0 = 2$pixel to simulate the real noise in the matching process.

The results obtained using the proposed algorithm are represented in Table 5.3, with the estimates for experiment 5 illustrated in Fig. 5.4 as an example for analysis.

To evaluate the system in terms of its accuracy we compared the estimates with the ground-truth values represented in Table 5.3. As we can see the error between the real and estimated offsets is very low regardless of the starting position of the head. In this case we have a maximum error of 7% (or 2.19deg) for the left eye pan joint in experiment 3. The eyes offsets are the ones presenting the largest errors (average error of 0.99deg) which can be explained by the approximation taken for the calibration system, where we are assuming the world is static and all points are represented at infinity which may

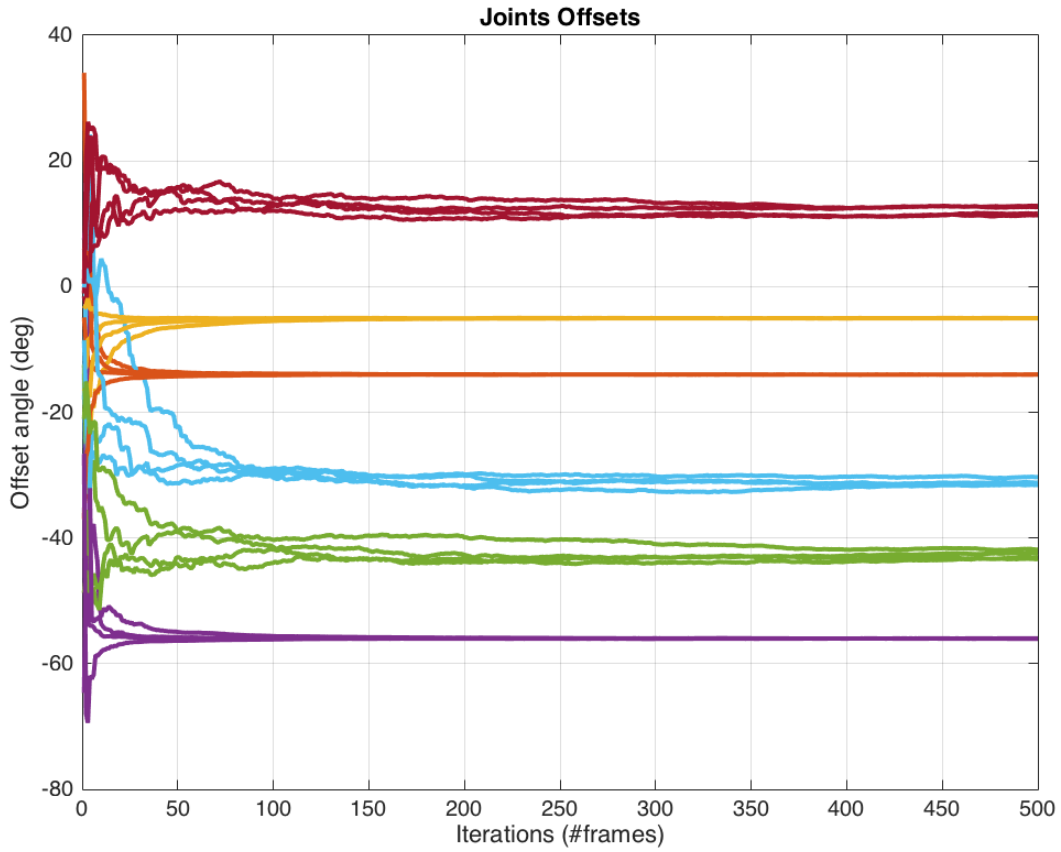


Figure 5.4: Simulated experiments: head calibration offsets estimates for experiment 5 (5 trials), with the ground-truth values represented in Table 5.3: Neck tilt δ_0 (in orange), Neck swing δ_1 (in yellow), Neck pan δ_2 (in purple), Eyes tilt δ_3 (in green), Left eye pan δ_4 (in cyan) and Right eye pan δ_5 (in red)

# Experiment	δ_0(deg)	δ_1(deg)	δ_2(deg)	δ_3(deg)	δ_4(deg)	δ_5(deg)
1 (ground-truth)	13.00	24.00	38.00	30.00	18.00	-9.00
1 (mean)	13.02	23.97	38.01	31.40	17.99	-7.27
1 (std)	0.01	0.01	0.02	0.29	0.19	0.16
1 (mean error)	0.02	0.02	0.01	1.40	0.01	1.72
2 (ground-truth)	-9.00	-11.00	-35.00	-39.00	-24.00	-9.00
2 (mean)	-9.01	-11.02	-34.98	-39.11	-22.49	-8.36
2 (std)	0.01	0.02	0.03	0.41	0.17	0.40
2 (mean error)	0.01	0.02	0.01	0.12	1.50	0.63
3 (ground-truth)	-41.00	-3.00	30.00	-8.00	-35.00	0.00
3 (mean)	-40.95	-3.01	30.00	-8.26	-32.80	0.40
3 (std)	0.01	0.01	0.02	0.30	0.25	0.38
3 (mean error)	0.04	0.01	0.01	0.26	2.19	0.40
4 (ground-truth)	-20.00	-6.00	19.00	18.00	29.00	4.00
4 (mean)	-19.97	-6.01	19.00	19.46	28.34	4.94
4 (std)	0.01	0.01	0.02	0.21	0.18	0.20
4 (mean error)	0.02	0.01	1.01	1.46	0.65	0.94
5 (ground-truth)	-14.00	-5.00	-56.00	-43.00	-33.00	12.00
5 (mean)	-13.99	-5.01	-56.01	-43.14	-31.55	11.67
5 (std)	0.02	0.01	0.02	0.38	0.27	0.46
5 (mean error)	0.00	0.01	0.00	0.14	1.44	0.32

Table 5.3: Simulated Experiments: ground-truth and statistical results of the head calibration offsets estimates for 5 experiments, with 5 trials each.

introduce parallax errors that are compensated by the system as errors in the joint space. All the other joints are able to converge to the correct values with much lower errors, thus proving the accuracy of the proposed calibration system.

By observing Fig. 5.4, we can see that the system converges in less than 150 iterations for all the 5 trials which, considering the lowest sensor runs at 30Hz and the system works in real time, corresponds to a convergence in less than 5s. This is very important since the robot can be rapidly calibrated before operation without consuming much of the operator's time. We can see each estimate remains almost constant after convergence, even if the head is continuously rotating. The low standard deviation values represented in Table 5.3 show the stability of the system, keeping the estimates as constant as possible under different operation conditions, while the head was being rotated. This is extremely important considering the robot is being calibrated in an online fashion and should keep a well calibrated internal model at all times.

5.2 Real Experiments

The validation of the proposed architecture was very important before testing it in the real robot. The quality of the previous results gave us confidence to test the calibration system in the real robot. Unfortunately, in the real case, there is no ground-truth for the real offsets of the robot, so we can only assess its repeatability.

We started by performing four experiments (experiments 1 to 4) where we initialized the robot head at different arbitrary positions, with a full reset of the encoders between each experiment. For each experiment we performed five trials where we randomly rotated the robot head and eyes by hand during 33.33 seconds (1000 iterations) so as to span most of the range of the robots joints. The mean and standard deviations for each experiment are represented in Table 5.4, with the estimates illustrated in Fig. 5.5 for experiment 4 as an example for analysis.

From Fig. 5.5 we can see that the system converged in less than 200 iterations to very similar estimates despite the different induced trajectories. These results show the capability of the system to correctly calibrate the offsets of the encoders regardless of their starting position, which is of utmost importance for any robotic platform. The low standard deviations for each experiments (maximum value of 0.85deg), observed in Table 5.4 also show the stability of the system, which keeps its estimates as constant as possible while the head was being rotated.

The accuracy of the head calibration system is hard to measure in the real robot, given the lack of ground-truth for each joint. Using the available sensors we measured

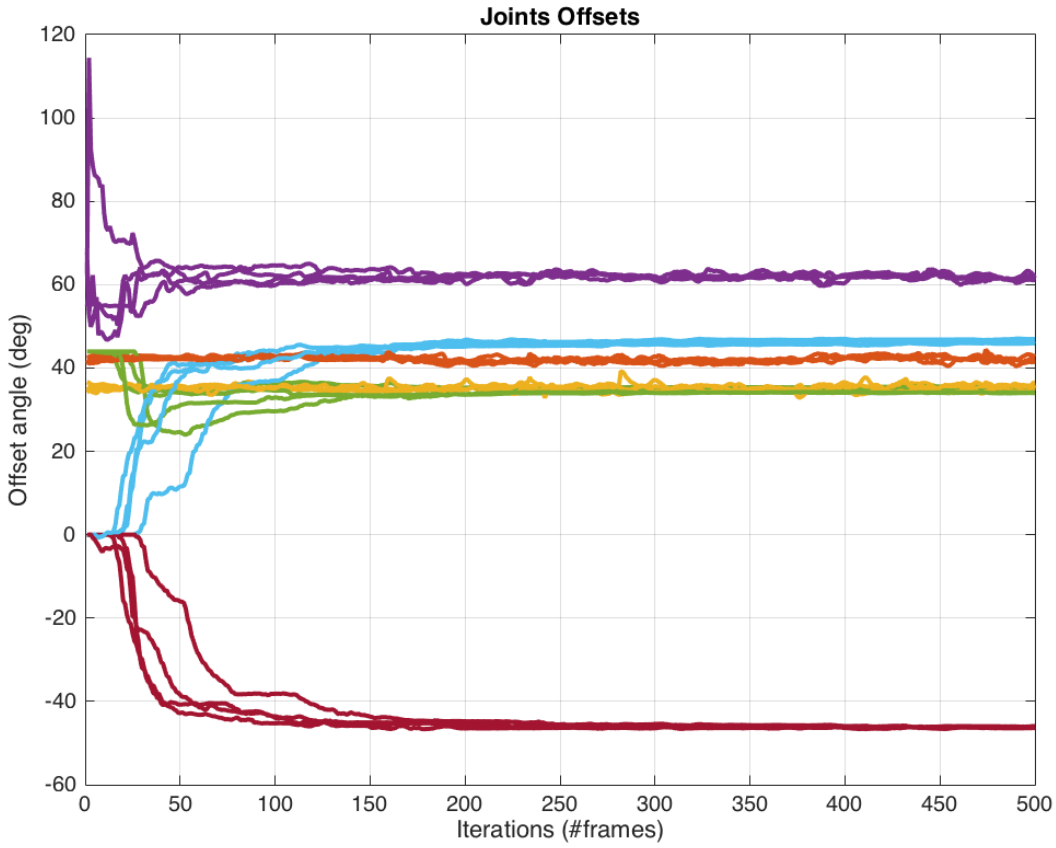


Figure 5.5: Real experiments: head calibration offsets estimates for experiment 4 (5 trials): Neck tilt δ_0 (in orange), Neck swing δ_1 (in yellow), Neck pan δ_2 (in purple), Eyes tilt δ_3 (in green), Left eye pan δ_4 (in cyan) and Right eye pan δ_5 (in red)

# Exp.	δ_0(deg)	δ_1(deg)	δ_2(deg)	δ_3(deg)	δ_4(deg)	δ_5(deg)
1 (mean)	-41.44	-41.13	62.52	3.49	-48.41	44.63
1 (std)	0.60	0.68	0.73	0.11	0.35	0.27
2 (mean)	-46.82	42.91	62.82	1.34	-49.22	-47.00
2 (std)	0.68	0.73	0.84	0.38	0.38	0.42
3 (mean)	42.87	11.41	-53.97	-36.17	-49.68	-46.13
3 (std)	0.70	0.68	0.77	0.25	0.19	0.21
4 (mean)	41.93	35.24	61.68	34.73	46.75	-46.19
4 (std)	0.64	0.68	0.85	0.49	0.34	0.36

Table 5.4: Real Experiments: mean and standard deviation values of the offsets estimates for all the experiments (5 trials for each experiment)

# Exp.	δ_0(deg)	δ_1(deg)	δ_2(deg)	δ_3(deg)	δ_4(deg)	δ_5(deg)
5	-47.20	23.93	-56.22	-17.53	-36.12	32.90
6	-34.14	-35.17	49.23	-19.21	38.83	-40.09
7	43.80	-25.82	50.91	-22.91	26.51	-41.99
8	43.39	28.47	-44.03	-21.15	23.43	-39.52

Table 5.5: Real Experiments: offsets estimates used to home the head to its zero position

# Experiment	$g_x(m/s^2)$	$g_y(m/s^2)$	$g_z(m/s^2)$
5	-0.022	-9.835	-0.016
6	0.034	-9.844	-0.096
7	-0.102	-9.829	-0.091
8	-0.051	-9.851	-0.010

Table 5.6: Real Experiments: gravity vector components in the zero position

the accuracy of the neck joints by comparing the real IMU's linear acceleration with the one predicted by the calibrated kinematic model of the robot head. We performed four experiments (experiment 5 to 8), with one trial per each, where we initialized the head at different arbitrary positions with a full reset of the encoders between each experiment. After calibration was achieved, for each case the robotic head was homed to the calibrated zero position, using the calibrated offsets to define the absolute zero position of the head. The calibrated offsets and the corresponding recorded gravity vector readings are shown in Table 5.5 and 5.6, respectively.

We can see that the gravity vector is practically vertical (with an accuracy of 99.99%) for all four experiments. These results demonstrate the ability of the system to converge to a solution which agrees with the absolute gravity readings, when started in completely different head configurations, thus correctly setting the absolute zero position of the head. After homing to the zero position, for each experiment, we applied a rectangular signal to the neck tilt joint (joint δ_0) in order to compare the real and predicted observations for the gravity vector. The comparison is represented in Fig. 5.6 for experiment 5 as an example for analysis.

Fig. 5.6 correctly shows a leaning forward and backwards pattern, after iteration 1700, corresponding to the signal applied to the first joint. After convergence of the filter, the predicted and real observations for the gravity vector are correctly aligned, with the prediction matching the real signal in more than 99.9% of the time. It is important to refer that the IMU is assumed to be perfectly mounted on the top of the

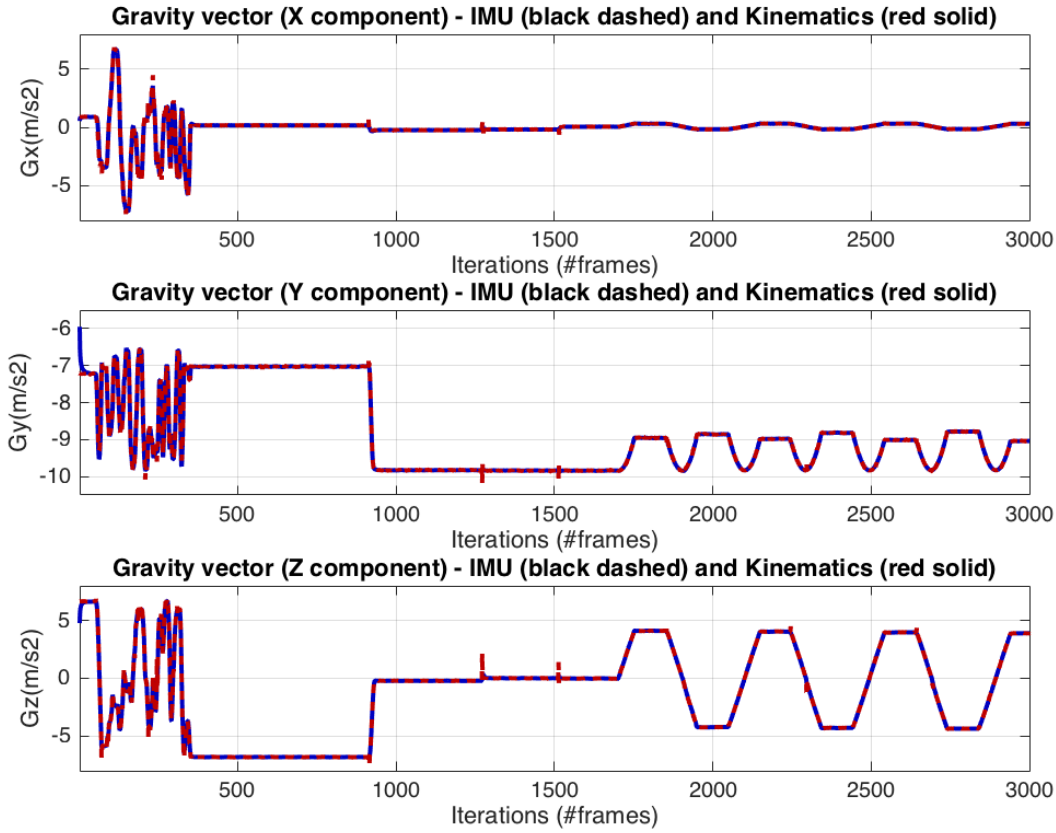


Figure 5.6: Real (red dashed) and predicted (blue solid) gravity vectors for Experiment 5, with homing to zero position after convergence and response to rectangular signal applied to the first joint, neck tilt.

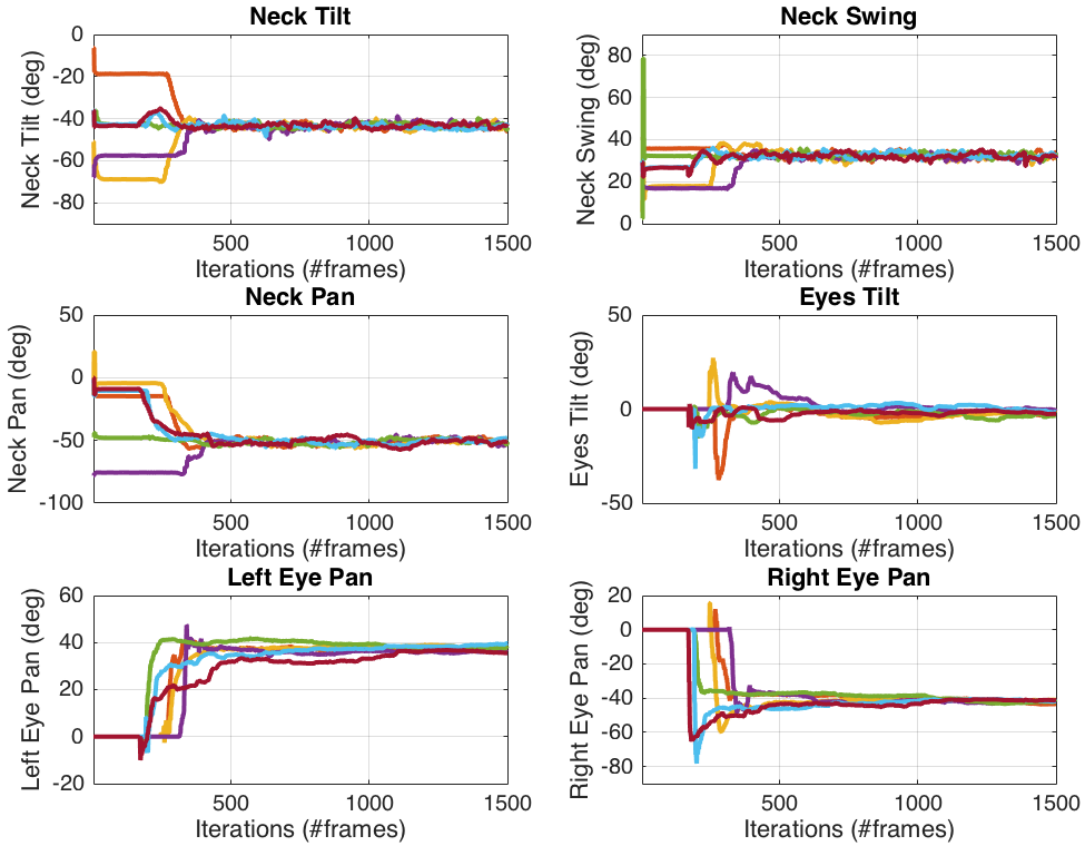


Figure 5.7: Repeatability: head calibration offsets estimates for experiment 9, with 6 trials, without a full reset of the encoders, showing the repeatability of the system

head, without any mounting error which may not be true. In that case, the mounting error of the sensor will be reflected in the offsets estimates in order to approximate the real and predicted IMU signals.

To better analyse the repeatability of the system, we performed experiment 9 where we ran the algorithm with the robot head started in six different configurations without a full reset of the encoders, meaning the offsets were the same for all trials. Fig. 5.7 shows the convergence of the offset estimates in each trial, with the mean value taken in the last 500 iterations shown in Table 5.7 along with the standard deviation of the estimates for all trials. The results show the estimates converging to similar values for different trials, thus empirically proving the robustness of the proposed method to very different starting conditions. The system's repeatability is extremely important to guarantee the quality of the filter, allowing the correct operation of the platform.

It is worth noting, in Fig. 5.7, that immediately after the first iteration the sys-

# Trial	δ_0(deg)	δ_1(deg)	δ_2(deg)	δ_3(deg)	δ_4(deg)	δ_5(deg)
1	-43.8	31.7	-50.2	-2.1	37.3	-42.4
2	-43.4	32.6	-52.7	-3.4	35.5	-43.7
3	-43.4	31.4	-50.9	-1.5	36.9	-40.4
4	-43.1	32.0	-52.0	-3.9	37.4	-42.2
5	-43.4	32.8	-51.1	-2.3	39.8	-42.1
6	-43.0	33.0	-52.9	-3.9	35.7	-41.6
mean	-43.57	32.31	-50.54	-2.72	36.85	-42.20
std	0.28	0.64	1.07	1.02	1.54	1.08

Table 5.7: Repatability: mean values of the offsets estimates for experiment 9, with 6 trials, without a full reset of the encoders

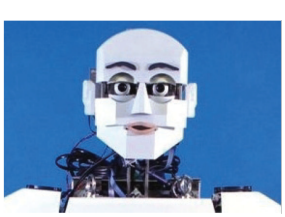
tem has already assimilated the first reading of the accelerometer. Since different head configurations can provide the same accelerometer readings (e.g. different pan angles with a fully upright head), the initial measurements are not enough to converge to the final configuration nor do they provide any information about the eye joints. The head movements are required to disambiguate these multiple solutions and provide the final offsets values, as seen by the rapid convergence of the filter after rotating the head.

During operation we noticed there were backlash zones in some of the joints, specially those carrying most of the weight. Within these backlash zones the encoders can not provide any measurements even though the joint is rotating in its motor shaft. However, the backlash was not reflected in the final estimates which shows that our system can adapt to sudden changes and perturbations that may occur during operation, mainly due to sensor fusion. The IMU and the cameras could perceive movement even though the encoders were telling the exact opposite. Sensor fusion is extremely useful to increase the robustness of the system in several situations where one or more sensors could fail. This case is a clear example of how multiple sensors integrated into a single architecture generate a better response than each one of them separated.

5.3 Generality

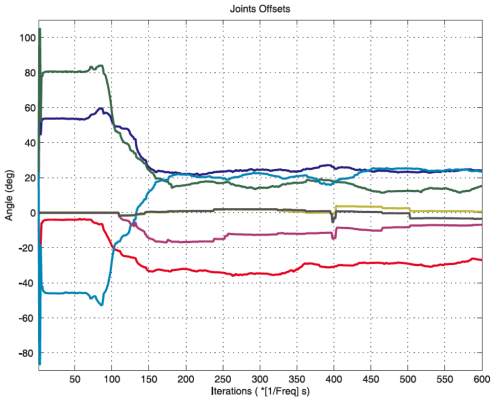
To show the generality of the head calibration system and how it could be applied to different kinematic models with similar sensors, we've implemented the system in two different robotic heads, the KOBIAN with 7 DOF [32], and the Vizzy with 6 DOF [33]. The calibration results are presented in Fig. 5.8.

The convergence of the system to the correct offsets values shows that the proposed

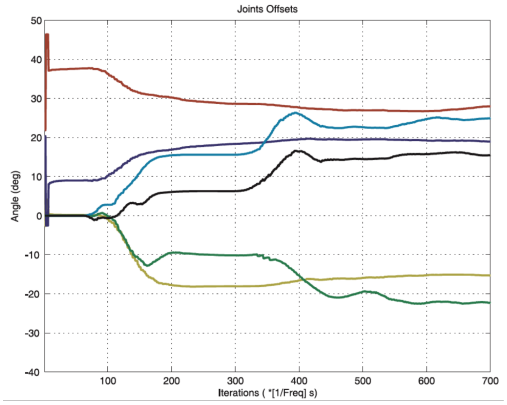


KOBIAN head

Vizzy head



a)



b)

Figure 5.8: Application of the head calibration procedure to different robot-heads, namely the KOBIAN head and the Vizzy head. Results for the KOBIAN joint offsets, joints 0 till 6, are coded in colors blue, green, red, light blue, purple, yellow and black. a) Offsets estimated along time for the seven joints of the KOBIAN and b) the six joints of the Vizzy.

calibration methodology can be applied to different robotic platforms regardless of the kinematic chain, when equipped with either image or IMU. This is of utmost importance for the robotic community given the high number of platforms with different kinematic configurations. The calibration system was also used in the European project RoboSoM to correctly calibrate the SABIAN robotic head, [14], during locomotion, which highlights the importance of such a system in real and complex robotic applications.

Chapter 6

Stereo Calibration System

In this chapter we will explain in detail the design and implementation of the proposed online stereo calibration system based on an Implicit Extended Kalman Filter. The proposed system can work with stereo cameras, in any possible configuration if the region of intersection between the two images is large enough to generate the pairs of matched features required for the system to correctly perform the calibration. In Section 6.1 we start by introducing an online stereo calibration system within a single filter architecture. Following an observability analysis presented in Section 6.2, that provides guidelines for the selection of which measurements are informative enough to estimate each parameter of the stereo system, we show the inability of the former system to perform measurements' selection within a single architecture. We then propose a multiple filter architecture in Section 6.3 that aims at solving the stereo calibration problem by using measurement selection, where we separated the stereo calibration system presented in Section 6.1 into five multiple sub-systems.

6.1 Classical Filter Architecture

The first implemented architecture consisted of a classical filter to estimate all the parameters at once. An IEKF is used since none of the measurement equations can be written in explicit form, defining instead a constraint that the measurements, together with the system state, need to satisfy.

6.1.1 State Transition Model

The adopted stereo model has only 5 DOF, with a fixed baseline, as stated in Section 2.2, where the t_x component is directly obtained from the constraint. Considering the

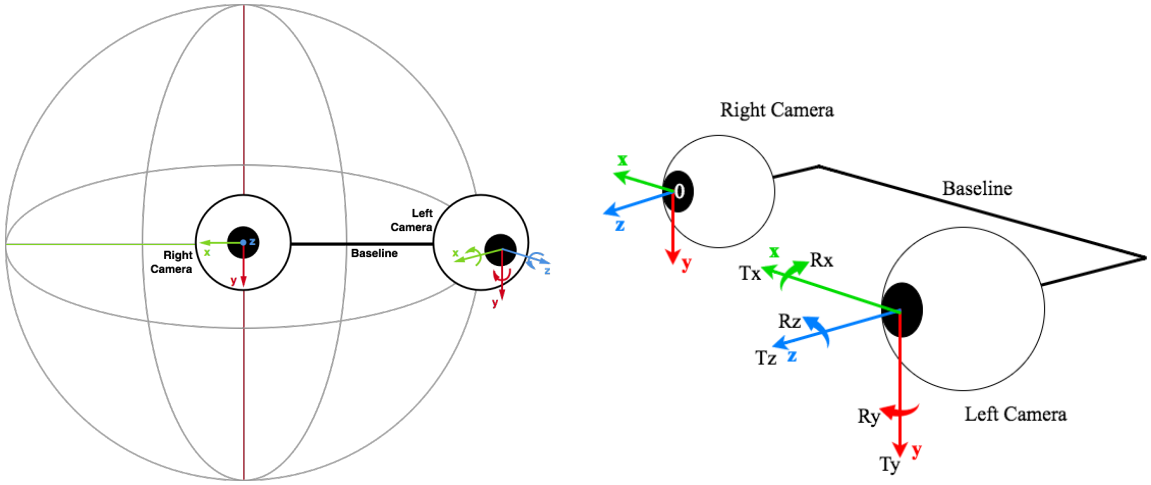


Figure 6.1: The adopted stereo model with 5 DOF; a) spherical representation of the stereo model where one of the cameras is fixed at the sphere's center and the other camera can freely move around it on the surface of the sphere (fixed baseline constraint); b) the stereo calibration parameters represented in the corresponding axis with the correct orientations.

proposed model the system state x_S is given by:

$$x_S = [t_y, t_z, r_x, r_y, r_z] \tag{6.1}$$

where t_y and t_z correspond to the translational components and r_x, r_y and r_z correspond to the axis-angle rotational components representing the group of parameters of the stereo roto-translation between the left and right cameras, as seen in Fig. 6.1. These parameters are assumed to be almost constant between two consecutive time instants, thus the state transition function f is the identity and the transition model simply propagates the previous values with some state transition noise w_S^k.

$$x_S^{k+1} = x_S^k + w_S^k \tag{6.2}$$

Here $w_S^k \sim \mathcal{N}\left(0, Q_S^k\right)$ where Q_S^k represents the covariance matrix of the zero mean state transition noise w_S^k, assumed Gaussian. The system can be adapted to be more or less responsive to variations in the translational and rotational components by changing this covariance matrix.

System Initialization

We want to estimate the roto-translation parameters of a stereo platform, with a known and fixed baseline. Therefore, the system state x_S^0 is initialized with the cameras at their nominal position with both optical axis pointing to the front in a perfectly parallel configuration, which corresponds to set all the parameters to 0:

$$x_S^0 = \mathbf{0}^5 \in \mathbb{R}^5 \tag{6.3}$$

The covariance matrices P_S^0 and Q_S^0, corresponding to the system state uncertainty and the system uncertainty during the state transition process respectively, are both diagonal. The P_S^0 matrix is initialized with the standard deviation values set for the system state uncertainty in the translational parameters, $\sigma_{Px^t}^0$, and in the rotational parameters, $\sigma_{Px^R}^0$ or our confidence on the initial system state values. The Q_S^0 matrix is also initialized with the standard deviation values set for the system process noise, again for the translational and rotational parameters, $\sigma_{Qx^t}^0$ and $\sigma_{Qx^R}^0$, or how we believe the system state variables will change between two consecutive time instants:

$$\begin{cases} P_S^0 = diag\left(\left(P_S^t\right)^0, \left(P_S^R\right)^0\right) \in \mathbb{R}^{5\times5} \\ Q_H^0 = diag\left(\left(Q_S^t\right)^0, \left(Q_S^R\right)^0\right) \in \mathbb{R}^{5\times5} \end{cases} \tag{6.4}$$

where

$$\begin{cases} \left(P_S^t\right)^0 = \boldsymbol{I}^2.\left(\sigma_{Px^t}^0\right)^2 \in \mathbb{R}^{2\times2} \\ \left(P_S^R\right)^0 = \boldsymbol{I}^3.\left(\sigma_{Px^R}^0\right)^2 \in \mathbb{R}^{3\times3} \\ \left(Q_S^t\right)^0 = \boldsymbol{I}^2.\left(\sigma_{Qx^t}^0\right)^2 \in \mathbb{R}^{2\times3} \\ \left(Q_S^R\right)^0 = \boldsymbol{I}^3.\left(\sigma_{Qx^R}^0\right)^2 \in \mathbb{R}^{3\times3} \end{cases} \tag{6.5}$$

where $\boldsymbol{I}^N$ corresponds to an N dimensional identity matrix and where we assume each translational parameter under estimation has the same level of uncertainty (analogous for the rotational parameters).

6.1.2 Observation Model

The cameras provide N pairs of matched image features between the left and right cameras, represented by their image coordinates $f_i = [u_i\,,\, v_i] \in \mathbb{R}^2$, collected in feature measurement vectors ${}^{L}y_F^{k+1} \in \mathbb{R}^{2N}$ and ${}^{R}y_F^{k+1} \in \mathbb{R}^{2N}$. The structure of ${}^{L}y_F^{k+1}$ (this is analogous for ${}^{R}y_F^{k+1}$) is given by:

$$^{L}y_F^{k+1} = \left[^{L}f_0^{k+1} \dots ^{L}f_{N-1}^{k+1} \right] \tag{6.6}$$

The system measurements y^{k+1} are, at each time instant $k+1$, given by:

$$y^{k+1} = \left[^{L}y_F^{k+1}, ^{R}y_F^{k+1} \right]^T + v_S^k \tag{6.7}$$

where $v_S^k \sim \mathcal{N}\left(0, R_S^k\right)$ is the observation noise assumed to be a zero mean Gaussian with covariance matrix $R_S^k \in \mathbb{R}^{4N \times 4N}$. These measurements provide geometrical constraints that are used by the filter to compute the state estimate.

The complete roto-translation $^{R}T_L$, from the left camera to the right camera, is obtained from the parameters in x_S and will be used to build the Fundamental Matrix, considering the filter's observation model uses the Fundamental Matrix Constraint as the cost functional to be minimized. In perfect conditions, if we take a pair of image features p_l and p_r, represented by their homogeneous image coordinates, the following relation holds:

$$p_r^T F(x_S) p_l = 0 \tag{6.8}$$

where F corresponds to the Fundamental Matrix previously described in equation (3.7). In cases where this relation does not hold, due to noise in the points coordinates or errors in the Fundamental Matrix, the constraint will not be zero, but some non-zero residue. This residue encodes the distance to the epipolar line but can not be used as a metric due to scale ambiguity in the representation of F. Let's consider the right epipolar line l_R:

$$l_R = F(x_S)p = \left(l_R^{(1)}, l_R^{(2)}, l_R^{(3)} \right)^T \tag{6.9}$$

Then, to obtain the distance in pixel units from point p_r to its corresponding epipolar line l_R we have to normalize the components of the epipolar line, [23], so that $l_R^* = l_R / \sqrt{l_R^{(1)} + l_R^{(2)}}$ and compute the epipolar distance $\epsilon_R = p_r^T l_R^*$ (the distance for the left epipolar line ϵ_L is computed the same way as ϵ_R). This signed distance tells us how far, in pixel units, points are from their corresponding epipolar lines, and if they are above or under the line. A quadratic distance can be computed by considering the epipolar distances in both images [17]:

$$\epsilon = \epsilon_R^2 + \epsilon_L^2 = 2\epsilon_R^2 \tag{6.10}$$

where we use the squared error instead of its normal value considering it improves the filter's convergence rate. This error can be written as: $\epsilon = 2\left(p_r^T F(x_S) p_l\right)^2$, where

the dependency of the fundamental matrix on the extrinsic parameters x_S is noted. Computing the jacobian of the error ϵ with respect to the parameters we get:

$$\frac{\partial \epsilon}{\partial x_S} = 4\epsilon_R \left(p_r^T \frac{\partial F(x_S)}{\partial x_S} p_l \right) \tag{6.11}$$

This jacobian contains the sign information in the ϵ_R term, which will be of utmost importance for the correct update of the stereo calibration filter. Using equation (6.10), we compute the distance ϵ and use it in our filter's observation function as an implicit constraint to estimate the roto-translation components between the two cameras. To accomplish this we apply equations (6.9) and (6.10) to each pair of features i represented in ${}^L y_F^{k+1}$ and ${}^R y_F^{k+1}$. From each pair of features i we obtain two distance measurements, ${}^L\epsilon_i$ and ${}^R\epsilon_i$, representing the distances from the left and right features to their corresponding epipolar lines, respectively. The system measurements constraint $\bar{z}_S^{k+1}$ is then given by:

$$\bar{z}_S^{k+1} = h_S \left(\bar{x_S}^{k+1}, {}^L y_F^{k+1}, {}^R y_F^{k+1} \right) = [\epsilon_0, \ldots, \epsilon_{N-1}]^T \tag{6.12}$$

where ϵ_i represents the quadratic epipolar distance for each point match i. The sign information from the distance to the epipolar lines is reflected in the jacobian matrix H_S evaluated at the predicted state value $\bar{x}_S^{k+1}$, given by:

$$H_S = \left. \frac{\partial h_s}{\partial x_S} \right|_{x_S = \bar{x_S}^{k+1}} = \left[\begin{array}{ccc} \frac{\partial \epsilon_0^k}{\partial x_S}^T & \ldots & \frac{\partial \epsilon_{N-1}^k}{\partial x_S}^T \end{array} \right]^T_{x_S = \bar{x_S}^{k+1}} \tag{6.13}$$

with the partial derivatives being calculated as in (6.11).

The observation noise of the implicit measurement constraint, $\tilde{v}_S^{k+1} \sim \mathcal{N}\left(0, \tilde{R}_S^{k+1}\right)$, is assumed to be a zero mean Gaussian with covariance matrix $\tilde{R}_S^{k+1}$ obtained using the equation described in (3.34). The observation covariance matrix R^{k+1}, required to calculate $\tilde{R}_S^{k+1}$, must be initialized using the standard deviation values for the image features, σ_{Rf}^0, or the uncertainty we have in the visual observations obtained from the pair of images. Considering we have M image features for each camera, the covariance matrix R^{k+1} is given by:

$$R^{k+1} = \boldsymbol{I}^{2M} \left(\sigma_{Rf}^0\right)^2 \in \mathbb{R}^{2M \times 2M} \tag{6.14}$$

where $\boldsymbol{I}^N$ corresponds to an N dimensional identity matrix.

The constraint in (6.12) represents the Implicit Kalman filter's innovation function, depending both on the system state prediction $\bar{x}_S^{k+1}$ and on the measurements ${}^L y_F^{k+1}$

and ${}^{R}y_{F}^{k+1}$. By using this function as the filter's innovation, the system is able to estimate the complete roto-translation that minimize the distance between each point to its corresponding epipolar line.

To better understand the stereo calibration system, we have created a pseudo-code simulating one step of the filtering process, from a total of I iterations for a generic stereo platform. The pseudo-code is implementing the stereo calibration system using all the information described in this chapter jointly with the standard IEKF equations defined in Chapter 3.

6.2 Observability Analysis

In the previous implementation we were assuming all measurements could be used to estimate each parameter. However, some measurements are useless to estimate certain parameters since they provide no information about the real state of the stereo setup. For instance, it is known that points at large distance from the cameras are not informative enough for computing translation parameters. Also, points in the center of the image are not able to discriminate between rotations and translations that induce similar image motion.

In this section we will present an observability analysis to identify which points are good to be used for each parameter separately. This analysis takes into consideration the epipolar geometry of the stereo problem. Given an initial estimate of the fundamental matrix, we can rectify the images such that they are close to a nominal configuration, with colinear x axes and parallel y and z axes, according to their reference frames represented in Fig. 2.2b), as explained in Section 3.3. If the transformation is exact, we have perfectly horizontal epipolar lines for each pair of matched points between the two cameras. For the nominal position, we can take equation (3.8) and write the Fundamental Matrix Constraint for any pair of points p and p', as:

$$p'^{T} F(B, 0, 0, 0, 0, 0) p = 0 \tag{6.15}$$

where B corresponds to the baseline between the two cameras. If we now apply a small perturbation δ to one of the parameters the constraint will not be zero for the same corresponding points but an error ϵ instead.

$$p'^{T} F(..., \delta, ...) p = \epsilon \tag{6.16}$$

This means the Fundamental Matrix F cannot correctly explain the real roto-translation

Algorithm 3 Stereo Calibration System (pseudo-code)

Define $\sigma^0_{Px^t}$, $\sigma^0_{Px^R}$, $\sigma^0_{Qx^t}$, $\sigma^0_{Qx^R}$ and σ_{Rf} ▷ Standard deviation values

Initialization

$x^0_S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

$\left(P^t_S\right)^0 = eye(3)(\sigma^0_{Px^t})^2$

$\left(P^R_S\right)^0 = eye(3)(\sigma^0_{Px^R})^2$

$P^0_S = diag(\left(P^t_S\right)^0, \left(P^R_S\right)^0)$ ▷ Eq.6.4

$\left(Q^t_S\right)^0 = eye(3)(\sigma^0_{Qx^t})^2$

$\left(Q^R_S\right)^0 = eye(3)(\sigma^0_{Qx^R})^2$

$Q^0_S = diag(\left(Q^t_S\right)^0, \left(Q^R_S\right)^0)$ ▷ Eq.6.4

$i = 0$

while $i < I$ **do**

Prediction Step

$\bar{x}^{i+1}_S = f_S\left(\hat{x}^i_S\right)$ ▷ Eq.6.38

$F^i_S = Jacobian\left(f_S, \hat{x}^i_S\right)$ ▷ Eq.3.36

$\bar{P}^{i+1}_S = F^i_S \hat{P}^i_S \left(F^i_S\right)^T + Q^0_S$

Update Step

$y^{i+1} = ImageFeatures\,()$

$R^{i+1} = eye(2M)(\sigma_{Rf})^2$

$\bar{z}^{i+1}_S = h_S\left(\bar{x}^{i+1}_S, y^{i+1}\right)$ ▷ Eq.6.12

$H^{i+1}_S = Jacobian\left(h_S, \bar{x}^{i+1}_S\right)$ ▷ Eq.3.36

$D^{i+1}_S = Jacobian\left(h_S, y^{i+1}\right)$ ▷ Eq.3.36

$\tilde{R}^{i+1} = D^{i+1}_S R^{i+1} \left(D^{i+1}_S\right)^T$

$K^{i+1} = \bar{P}^{i+1}_S H^{i+1}_S \left(H^{i+1}_S \bar{P}^{i+1}_S \left(H^{i+1}_S\right)^T + \tilde{R}^{i+1}\right)^{-1}$

$\hat{x}^{i+1}_S = \bar{x}^{i+1}_S - K^{i+1}\bar{z}^{i+1}_S$

$\hat{P}^{i+1}_S = \left(I - K^{i+1}H^{i+1}_S\right)\bar{P}^{i+1}_S \left(I - K^{i+1}H^{i+1}_S\right)^T + K^{i+1}\tilde{R}^{i+1}\left(K^{i+1}\right)^T$

$i = i + 1$

end while

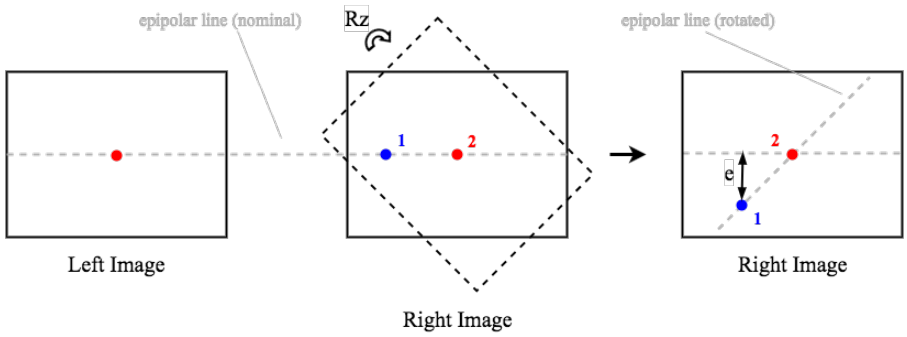


Figure 6.2: Example of lack of observability for a rotation r_z: Point 1 (in blue, near the left margin of the image) generates a large vertical displacement ϵ to the nominal epipolar line which helps to explain the rotation applied to the right image. Point 2 (in red, in the center of the image) has no vertical displacement and any applied rotation would generate a zero error, which makes it impossible to perceive and estimate the real rotation r_z.

between the two cameras. The value of ϵ will depend not only on which parameter was perturbed but also on the location of the corresponding points in the image. Let's consider the example in Fig. 6.2.

Starting from a configuration with parallel cameras, we acquire two observations represented as point 1 (blue) and point 2 (red), in the right image. We then apply a perturbation in r_z to the right camera only and acquire the new coordinates of the points. While the blue point lies on top of the rotated epipolar line and has a large distance ϵ to the nominal epipolar line, the red point lies exactly on top of both epipolar lines. The blue point gives much more information about this rotation than the red point. The latter cannot explain the rotation at all. This is an observability problem and we can easily see that not all points are good to estimate a particular parameter. The signal to noise ratio (SNR) of the observations around the red point will be very low for the purpose of estimating r_z. Any rotation observed with such points will give severely wrong values. However, those points may be good to estimate any other parameter so they should not be fully discarded, but selected purposefully for the estimation of parameters for which they yield a good SNR. The careful selection of the observations can prevent the introduction of noise in the estimation filter that may lead to parameter drift.

The proposed observability analysis takes into consideration the epipolar geometry of the stereo problem at the canonical stereo configuration. At this configuration any horizontal disparity d_h is related to the point's depth and a vertical disparity d_v is the

pixel epipolar error that reflects measurement noise or error in the extrinsic parameters. Thus, the proposed observability analysis will verify which vertical disparity d_v is produced in a certain image point if a variation δ is applied to a certain parameter. If the produced d_v is above a certain quantization or noise level $\mathcal{E}$, than a variation of δ in that parameter can be observed. Otherwise, the observed disparity in not informative about variations on that parameter. The value of d_v will depend not only on which parameter was perturbed but also on the location of the corresponding points in the image. The proposed observability analysis will allow us to select point matches based on the image locations that provide best observability for a certain parameter.

Let us first compute the general form of the vertical disparity, for perturbations around the canonical stereo configuration. According to the parametrization in (2.4) and (2.5), a world point $P_L = [X_L, Y_L, Z_L]$ in the left camera frame has the following vertical and depth coordinates in the right camera frame:

$$\begin{aligned} Y_R &= s_z c_y X_L + (c_z c_x + s_x s_y s_z) Y_L + (c_x s_y s_z - s_x c_z) Z_L + t_y \\ Z_R &= -s_y X_L + c_y s_x Y_L + c_y c_x Z_L + t_z \end{aligned} \tag{6.17}$$

For an arbitrary set of parameters, the vertical disparity is given by:

$$d_v = v_r - v_l = f_y (Y_R/Z_R - Y_L/Z_L) \tag{6.18}$$

Computing the gradient of d_v with respect to the vector of parameters θ around the canonical configuration, $\theta = 0$ we have:

$$\left[\frac{\partial d_v}{\partial \theta}\right]_{\theta=0} = f_y \begin{bmatrix} \frac{1}{Z_L} & \frac{-y_L}{Z_L} & -1-y_L^2 & x_L y_L & x_L \end{bmatrix} \tag{6.19}$$

Then, for small perturbations of the parameters around the canonical configuration we have:

$$d_v \approx \frac{f_y}{Z_L} t_y - \frac{f_y y_L}{Z_L} t_z - f_y (1 + y_L^2) r_x + f_y x_L y_L r_y + f_y x_L r_z \tag{6.20}$$

with $x_L = X_L/Z_L$ and $y_L = Y_L/Z_L$. This is the first-order approximation of the closed form solution presented in Appendix A, valid for small perturbations.

We will now analyse the effect of individual perturbations on each parameter and check an observability condition: the resulting vertical disparity must have an amplitude above the noise threshold $\mathcal{E}$. For a given perturbation amplitude δ, if the absolute value of the vertical disparity at a certain point is larger than $\mathcal{E}$, we say the parameter is observable from that point. To support the analysis some simulations will be performed

using images of 200×150pixel with optical centers $c_x = 100$pixel and $c_y = 75$pixel, focal lengths $f_x = f_y = 50$pixel and a baseline $B = 67$mm.

6.2.1 Observability of translational parameter t_y

A perturbation δ on the parameter t_y will result in the following vertical disparity:

$$d_v \approx \frac{f_y}{Z_L}\delta \tag{6.21}$$

The observability condition is verified if:

$$|d_v| > \mathcal{E} \quad \Longrightarrow \; Z_L < \frac{f_y\,|\delta|}{\mathcal{E}} \tag{6.22}$$

Considering the case $\mathcal{E} = 1$ corresponding to the minimum observable variation given the pixel discretization, we can see that points at distances Z_L larger than $f_y\,|\delta|$ will not generate any vertical disparity. This is an observability problem since the system can not differentiate between two different values for t_y that differ by an amount smaller or equal to δ. Only closer points can provide a large variation thus increasing the observability of the system, allowing the estimation of this parameter with an error lower than δ.

6.2.2 Observability of translational parameter t_z

A similar analysis can be done for the parameter t_z. The vertical disparity induced by a perturbation δ on this parameter is:

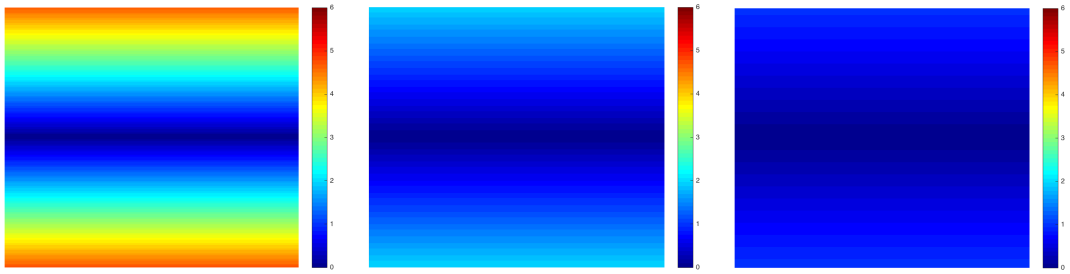
$$d_v \approx -f_y \frac{y_L}{Z_L}\delta \tag{6.23}$$

Fig. 6.3 displays d_v (rounded to integer values) induced by a perturbation $\delta = 6.7mm$, for different depths (but constant across the image). We can observe that larger depths require points that project closer to bottom or top of the image (large $|y_L|$).

The observability condition can be written as:

$$|d_v| > \mathcal{E} \implies Z_L < \frac{f_y|y_L|}{\mathcal{E}}\,|\delta| \tag{6.24}$$

Let us consider the case ($\mathcal{E} = 1$) and $\delta = 6.7$mm (10% of the baseline B in the given example). For points located at 70% (105pixel) and 90% (135pixel) of the image height this equation tells us that the maximum depth the point could take before the parameter loses its observability is 194.3mm and 395.3mm, respectively.



a) $z^L = 100$mm b) $z^L = 250$mm c) $z^L = 500$mm

Figure 6.3: Vertical disparity d_v when a small variation $\delta = 6.7$mm is applied to t_z (10% of the baseline) for a 200×150pixel image with optical centers $c_x = 100$pixel and $c_y = 75$pixel, focal lengths $f_x = f_y = 50$pixel and a stereo system baseline $B = 67$mm. Points within the same region (same color) will generate the same vertical variation (rounded to integer pixel units).

Both translational parameters t_y and t_z are very sensitive to the depth of the points used as measurements. Points at larger depths can not be used to correctly calibrate the translational components of a stereo system. Even though points near the top and bottom borders can compensate for larger depths, closer points are always preferable.

In case of the rotational parameters the analysis does not depend on the point's depth but on the spatial distribution of the points along the image. To maximize the information obtained from the measurements we must consider other aspects rather than the points depth, as we will see in the following subsections.

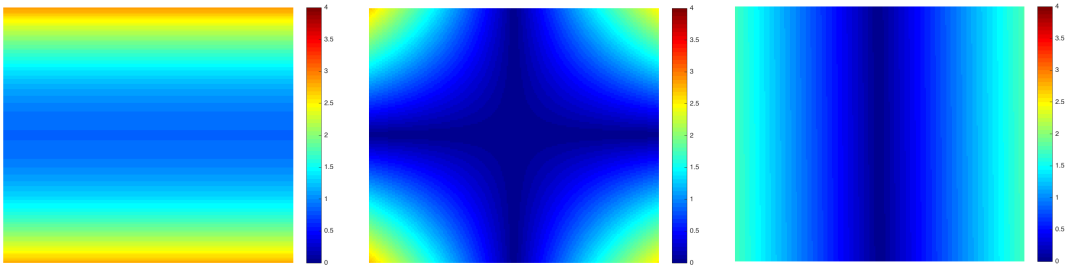
6.2.3 Observability of rotational parameter r_x

A perturbation of amplitude δ in the rotational parameter r_x influences the vertical the vertical disparity in the following way:

$$d_v \approx -f_y(1 + y_L^2)\delta \tag{6.25}$$

We see that points closer to the top and bottom borders of the image will provide better observability of r_x. This is illustrated with a simulation, in Fig. 6.4a), showing the amplitude of the vertical disparity at all image locations for a small rotation δ applied to r_x (1deg). The observability condition can be written as:

$$|d_v| > \mathcal{E} \implies y_L > \sqrt{\frac{\mathcal{E}}{f_y|\delta|} - 1} \quad \text{or} \quad y_L < -\sqrt{\frac{\mathcal{E}}{f_y|\delta|} - 1} \tag{6.26}$$



a) Rotation r_x b) Rotation r_y c) Rotation r_z

Figure 6.4: Vertical variation d_v of the points when a small rotation (1deg) is applied for a 200×150pixel image with optical centers $c_x = 100$pixel and $c_y = 75$pixel, focal lengths $f_x = f_y = 50$pixel and a baseline $B = 67$mm. Points within the same region (same color) will generate the same vertical variation, rounded to integer pixel units.

For example, with $\mathcal{E} = 1$, the observability boundaries are at $y_L = \pm 0.3780$ which correspond the vertical pixel coordinates $v_L = 93.9$ and $v_L = 56.1$. This means a variation greater than 1pixel in d_v can only be observed in image points coordinate v^L greater than 94pixel or lower than 56pixel.

6.2.4 Observability of rotational parameter r_y

For a perturbation δ in the rotational parameter r_y we have:

$$d_v \approx f_y x_L y_L \delta \tag{6.27}$$

The observability condition $|d_v| > \mathcal{E}$ can be expressed in terms of y_L:

$$y_L > \frac{\mathcal{E}}{f_y |x_L| |\delta|} \quad \text{or} \quad y_L < -\frac{\mathcal{E}}{f_y |x_L| |\delta|} \tag{6.28}$$

or in terms of x_L:

$$x_L > \frac{\mathcal{E}}{f_y |y_L| |\delta|} \quad \text{or} \quad x_L < -\frac{\mathcal{E}}{f_y |y_L| |\delta|} \tag{6.29}$$

Better observability will occur for points near the four corners of the image. Fig. 6.4b) shows for a simulation with $delta = 1$deg), highlighting the regions of larger variations. This figure shows the highlighted corners of the image where it is possible to get larger variations of d_v. At the image top ($v^L = 0$, $y^L = -1.5$), the x_L boundaries to have at least one pixel variation are $x_L = \pm 0.7619$. This means points at coordinates $(u_L > 138, v_L = 0)$ or $(u_L < 62, v_L = 0)$ will be sensitive to a rotation r_y of one degree, under 1 pixel discretization noise.

6.2.5 Observability of rotational parameter r_z

Let's consider now a variation δ applied to r_z. The vertical disparity changes as:

$$d_v \approx f_y x_L \delta \tag{6.30}$$

The observability condition $|d_v| > \mathcal{E}$ is now:

$$x_L > \frac{\mathcal{E}}{f_y|\delta|} \quad \text{or} \quad x_L < -\frac{\mathcal{E}}{f_y|\delta|} \tag{6.31}$$

We conclude that points that are closer to the left and right borders of the image result in better observability for parameter r_z. Fig. 6.4c) show a simulation for $\delta = 1$deg. Considering $\mathcal{E} = 1$ the boundaries for x^L are set to approximately 1.14 (right boundary) and -1.14 (left boundary), which, in pixel coordinates corresponds to u^L greater than 157pixel or lower than 42pixel.

Summary

This analysis gave us valuable information about the best measurements for each parameter under estimation. From the observability analysis, we have defined a set of optimal observations for each parameter that can be summarized as:

- t_y requiring points with a low depth, close to camera;
- t_z requiring points with a low depth, close to the camera, preferable those located near the top and bottom margins of the image;
- r_x requiring image points located near the top and bottom of the image;
- r_y requiring points located near the four corners of the image;
- r_z requiring points located near the left and right margins of the image

As already mentioned, this analysis approximates the vertical disparity d_v by its gradient, valid for small perturbations δ. The full and closed form solutions for each case are presented in detail in Appendix A, confirming the approximations presented in this section.

Other factors affecting the system's observability

The cameras' resolution is an important factor to take into account, as well as the lens used in our cameras (that set the focal length). The image discretization is responsible

for the loss of precious information relative to the coordinates of the image points. Even though translations in t_y for example, could generate small variations in d_v in a continuous space, in the discrete space those small changes may be lost and the vertical displacement may actually remain null. In terms of the system observability this means that the translation in t_y is non-observable since the points remain in the exact same location when in fact there was a clear variation in the y axis.

The higher the resolution the closest to a "continuous" space the image grid is. However larger images result in larger processing times, specially when performing feature matching. A compromise between speed and accuracy should be taken in order to get the best results in the required time.

Another important factor is the baseline which has a direct impact in the observability of the system since it allows the use of points at larger depths to be used as good measurements specially when estimating the translational components of the roto-translation. By analysing the above expressions we can see how the baseline affects the image coordinate variation d_v, specially in the cases of t_y and t_z. Larger baselines are good for systems working in areas where most of the world points are at a large distance. On the other hand for such large baselines we loose the ability to detect closer points since there is no intersection between the two images for such close regions.

Although these two factors are extremely important and can be tuned to provide the best results in terms of stereo calibration, they must not be considered as a solution to solve or improve the stereo calibration problem since they are a specification of the application running the system and not of the system itself. If the specifications for a certain application require the use of a small baseline, a larger baseline is not an option. For this reason we did the mathematical analysis of the system's observability which can adapt and guarantee the best results for all specifications in terms of image resolution, focal length and baseline.

6.3 Multiple Filter Architecture

The stereo observability analysis gave us valuable information about the stereo system that was not being taken into account in the classical filter architecture. Not all measurements should be treated equally. Some measurements are better for a specific component and provide more information than others. Having all the parameters under estimation in a classical filter makes it impossible to select or reject measurements since they are affecting each parameter in a different way. We could either feed the system with an observation or remove it from the group of all observations and not feed it at all. There

is not a way of differently weighting the observations for each parameter, thus informing the filter about their quality. One can say the Kalman gain K is already weighting the observations for each parameter, which is in part true but has some inherent problems. Let's consider a generic filter where we want to estimate two variables, x_1 and x_2. Let's assume now that we have only one observation, for which x_1 is non-observable (the innovation's output is the same for any value of x_1).

The Kalman gain equation, is given by:

$$K = \bar{P}H^T \left(H\bar{P}H^T + R\right)^{-1} \tag{6.32}$$

where, in this example, the jacobian H, for the observation function h, is given by:

$$H = \begin{bmatrix} \frac{\delta h}{\delta x_1} & \frac{\delta h}{\delta x_2} \end{bmatrix} \tag{6.33}$$

and the predicted covariance matrix $\bar{P}$ is given by:

$$\bar{P} = \begin{bmatrix} P_{x_1x_1} & P_{x_1x_2} \\ P_{x_2x_1} & P_{x_2x_2} \end{bmatrix} \tag{6.34}$$

In this example, for simplification purposes, we will assume there is no noise for the observations ($R = 0$). From this, the Kalman gain becomes:

$$K = \begin{bmatrix} \left(\frac{\delta h}{\delta x_1}P_{x_1x_1} + \frac{\delta h}{\delta x_2}P_{x_1x_2}\right)/A \\ \left(\frac{\delta h}{\delta x_1}P_{x_2x_1} + \frac{\delta h}{\delta x_2}P_{x_2x_2}\right)/A \end{bmatrix} \tag{6.35}$$

where A is given by:

$$A = \frac{\delta h}{\delta x_1}\left(\frac{\delta h}{\delta x_1}P_{x_1x_1} + \frac{\delta h}{\delta x_2}P_{x_2x_1}\right) + \frac{\delta h}{\delta x_2}\left(\frac{\delta h}{\delta x_1}P_{x_1x_2} + \frac{\delta h}{\delta x_2}P_{x_2x_2}\right) \tag{6.36}$$

Since x_1 is non-observable for the single observation, the innovation's output will be the same for any value of x_1. For this reason, the partial derivative of the jacobian H, $\delta h/\delta x_1$, will be zero. Thus, equation (6.35) can be simplified:

$$K = \begin{bmatrix} P_{x_1x_2}/\left(\frac{\delta h}{\delta x_2}P_{x_2x_2}\right) \\ 1/\frac{\delta h}{\delta x_2} \end{bmatrix} \tag{6.37}$$

As we can see, the Kalman gain for x_1 is not zero, even though the system is non-observable for this parameter. Considering the system can not observe any variation

in this parameter for its single observation, it should not update its estimate but keep it as it is, until another observation, hopefully with more information, arrives. The parameter's update is affected by a factor of $P_{x_1x_2}/P_{x_2x_2}$ which is small but not zero. In more complex systems such as ours, having more parameters under estimation and many more observations, we may experience high oscillations or even the divergence of some non-observable parameters when their estimates are being updated using only the contributions from other parameters. In this particular example, x_1 may start to slowly diverge until a new observation arrives that leads to $\delta h/\delta x_1 \neq 0$. Only this will allow the correct update of the parameter.

We know that for a certain observation, parameter x_1 is not observable yet we are still updating its estimate value. The fact that we could not control how each observation would affect the parameters was the main reason we decided to separate the classic filter architecture into multiple filters. Having multiple filters, one for each parameter, allowed us to carefully select which observations were used for each one, ensuring the parameters' estimates were only updated if relevant information was present.

We separated the main system in five sub-systems, implementing one filter per parameter under estimation, as seen in figure 6.1. The basis of each filter is also an IEKF but we are using the estimates of each filter as measurements for all the other filters. The following section explains in detail how each estimate works as a measurements for all the other sub-systems.

6.3.1 System Methodology

Each system state x_S we want to estimate is composed by only 1 parameter, either a translational parameter t_y or t_z or a rotational parameter r_x, r_y or r_z. To better understand this structure, the parameter under estimation will be named ${}^i x_S$, with $i = 1, ..., 5$ considering we have 5 parameters.

Dynamic Model

Each parameter is assumed to be approximately constant between two consecutive time instants. The transition model simply propagates the previous values with some state noise w_S^k:

$$ {}^i x_S^{k+1} = {}^i x_S^k + w_S^k \tag{6.38}$$

with $w_S^k \sim \mathcal{N}\left(0, q_S^2(k)\right)$. The system can be adapted to be more or less responsive to variations in the parameters by changing the variance q_S^2.

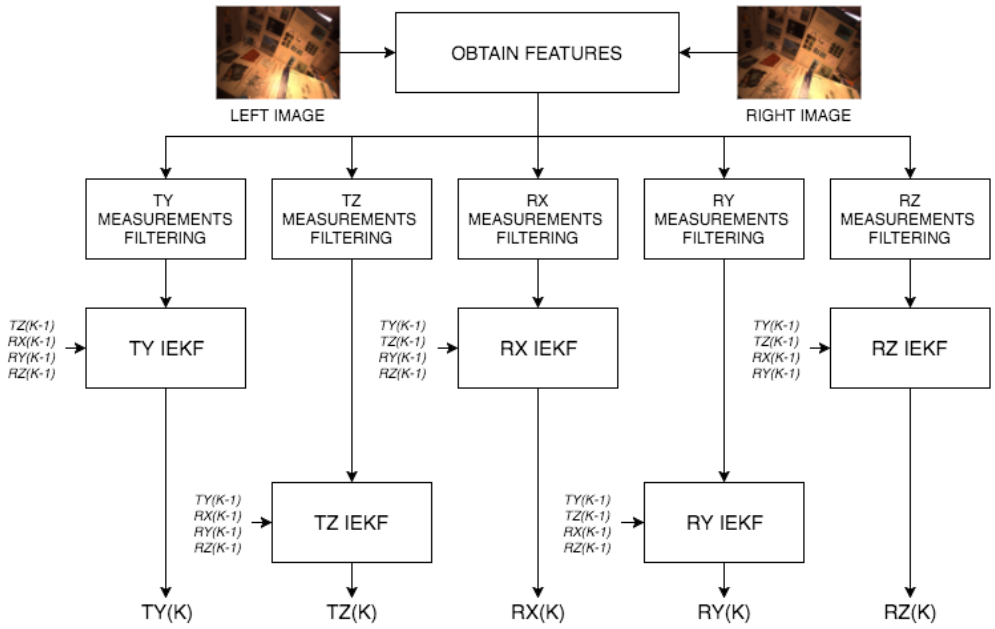


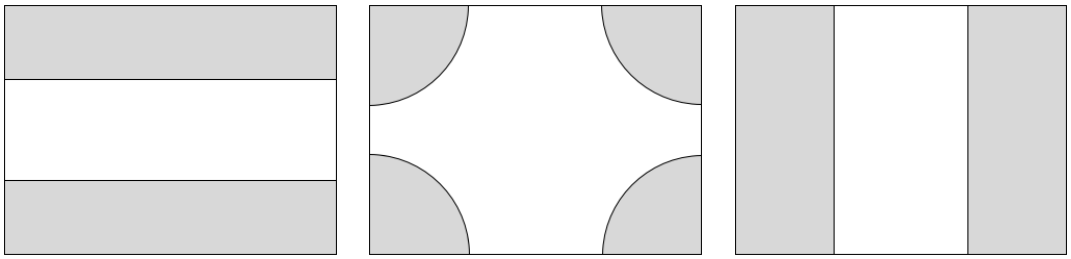
Table 6.1: Proposed architecture with a sub-system for each parameter under estimation. For each case the image features are filtered (selected using the previous analysis) to feed each sub-system with the best measurements only.

Observation Model

As in the single filter architecture, the cameras provide N pairs of matched image features between the left and right cameras. From the observability analysis, we select which measurements are better to update each parameter ${}^{i}x_S$ and collect them in vectors $\left({}^{L}y_F^{k+1}\right)_i \in \mathbb{R}^{2N_i}$ and $\left({}^{R}y_F^{k+1}\right)_i \in \mathbb{R}^{2N_i}$. For the translational parameters t_y and t_z, observation selection is based on point depth Z. Using stereo coordinates rectified to the canonical configuration (Section 3.3), the depth Z of the points can be obtained from eq. (3.13).

Let us consider a setup with 640×480 resolution cameras, intrinsic parameters $f_x = f_y = 340$, $c_x = 320$, $c_y = 240$, a baseline of $67mm$, and assume a measurement noise of 1 pixel, $\mathcal{E} = 1$. For a resolution δ of 5mm in the translational paramenters, according to (6.22) and (6.24) we can only acquire points up to a depth Z of 1700mm, for parameter t_y, and 1195mm for parameter t_z. This corresponds to a minimum horizontal disparity d of 13.4pixel and 19pixel, for t_y and t_z, respectively.

In case of the rotational parameters the best points do not depend on depth but solely on the image coordinates. Let us assume a resolution δ of 0.5deg for the rotational parameters. For r_x, and for the particular case considered here, (6.26) allows the



a) b) c)

Figure 6.5: Selected observations for the rotational filters (represented as the grey regions): a) Selected observations for r_x (42.8% from the top and bottom image borders for $\mathcal{E} = 3$); b) Selected observations for r_y (24% from the four image corners for $\mathcal{E} = 1$); c) Selected observations for r_z (31.8% from the left and right image borders for $\mathcal{E} = 1$)

acquisition of points from the entire image. If the noise threshold is higher (e.g. $\mathcal{E} = 3$) then only points close to the horizontal boundaries (placed at 42.5% from the top and bottom margins of the image) would be selected, as in Fig. 6.5a). For parameter r_y, using equations (6.28) and (6.29) we set the boundaries to 24.5% from each image corner as in Fig. 6.5b). Finally, in the case of parameter r_z, using equation (6.31), we set the boundaries to 32.1% from the left and right margins of the image, as in Fig. 6.5c).

After feature selection, the system measurements vector y_i^{k+1} is, at each time instant $k+1$, given by:

$$y_i^{k+1} = \left[{}^{j}x_S^{k+1}, \left({}^{L}y_F^{k+1} \right)_i, \left({}^{R}y_F^{k+1} \right)_i \right] + v_S^k \tag{6.39}$$

where $v_S^k \sim \mathcal{N}\left(0, R_S^k\right)$ is the measurement noise assumed to be a zero mean Gaussian with covariance matrix R_S^k, that includes both the observation noise and the estimation noise for each parameter, given by their covariance matrices; ${}^{j}x_S^{k+1} \in \mathbb{R}^4$, with $j = 1, ..., 5$ and $j \neq i$ corresponds to the current estimates of all the other parameters, and $\left({}^{L}y_F^{k+1} \right)_i$ corresponds to the left features selected for parameter i (analogous for the right features).

As in the single filter's case, these measurements, together with an estimate of the Fundamental matrix obtained from each current state prediction ${}^{j}\bar{x}_S^{k+1}$, are used to compute the innovation vector:

$$\bar{z}_S^{k+1} = \left[\begin{array}{ccc} (\epsilon_0)_i & \dots & (\epsilon_{N-1})_i \end{array} \right]^T \tag{6.40}$$

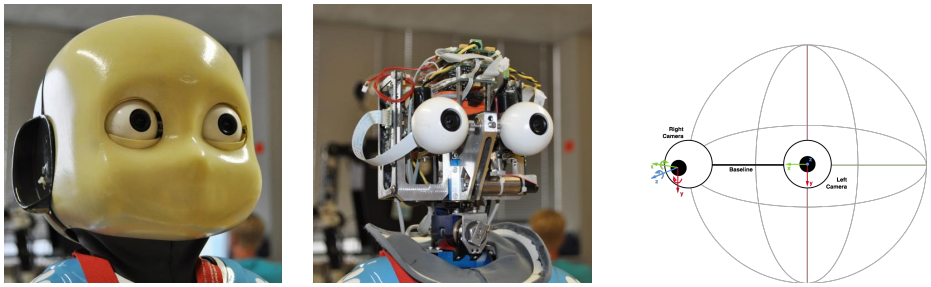
where the quadratic epipolar distance for each point match j, for parameter i, $(\epsilon_j)_i$ is computed as in (6.10). Again, the sign information from the distance to the epipolar lines is reflected in the Jacobian matrix $(H_S)_i$ evaluated at the predicted state value ${}^{j}\bar{x}_S^{k+1}$ for each parameter i.

Chapter 7

Stereo Calibration Results

In this chapter we will evaluate the proposed architecture for the stereo calibration system. We will perform simulated and real experiments to evaluate the system in terms of its accuracy, repeatability and robustness. The real experiments were performed using the same robotic platform as in the head calibration case, the iCub head prototype, named Chica.

To validate the proposed architecture with one filter per parameter as described in section 6.3 and demonstrate its advantage when compared to the first architecture with a classical filter, described in section 6.1, we performed simulated and real experiments where in the simulated case the cameras measurements (image features) were virtually generated and fed to the system just like in the real case. For both cases, the real and simulated, multiple datasets were created with ground-truth information. In the real case the ground-truth was obtained by acquiring calibration images with a chessboard pattern. The Bouguet Toolbox, [3] was then used to obtain the extrinsic parameters between the two cameras with a very low error, even though it is a very time consum-



a) Chica head b) Head structure c) Stereo kinematic model

Figure 7.1: The iCub robotic head used in our real stereo calibration experiments.

ing process. The cameras' intrinsic parameters were obtained using also the Bouguet Toolbox and are represented in Table 5.1.

The system's state uncertainty was initialized considering the stereo model, the initial configuration and the range of motions the stereo platform could take. At the beginning, the system state was initialized by setting the cameras at their nominal position, with perfectly parallel optical axis which corresponds to having $t_x = 1$ and $t_y = t_z = r_x = r_y = r_z = 0$, as seen in subsection 6.1.1. Therefore the system's state uncertainty should have large values for the standard deviations to better adapt to any initial configuration of the cameras. The uncertainty in the translational parameters $\sigma^0_{Px^t}$ and in the rotational parameters $\sigma^0_{Px^R}$ were initialized as 0.33 (dimensionless with respect to baseline length) and 20deg, respectively. This system assumes a slow drift, thus the transition process noises for the translational parameters $\sigma^0_{Qx^t}$ and for the rotational parameters $\sigma^0_{Qx^R}$ were set to 0.035 (dimensionless with respect to baseline length) and 0.5deg, respectively.

This system uses SIFT features as measurements, as explained in subsection 3.5.3 to find matches between two images, acquired at the exact same time instant. The noise level associated to these features measurements, σ^0_{Rf}, was assumed to be zero mean Gaussian noise with a standard deviation of 1pixel due to the high precision of the SIFT matching algorithm. Even though our cameras work at 30Hz, the SIFT extraction and matching is computationally heavy, limiting our calibration system to run at $\sim$ 15Hz. In the case of simulated experiments, instead of SIFT features we used virtual points, 3D points sampled from a Uniform Distribution. The image features used as measurements were then obtained by projecting each virtual point into the images. We added Zero-mean Gaussian noise to simulate the SIFT matching error, sampled from normal distribution with a standard deviation of 1pixel.

The stereo calibration system was designed to calibrate any pair of cameras with known intrinsic parameters, using the stereo model represented in figure 7c). The calibration procedure is very simple consisting in turning on the cameras and moving them around, using natural information from the environment. An example of the calibration procedure can be seen in this video: `https://www.youtube.com/watch?v=qbX6K92FCHk`

7.1 Validation

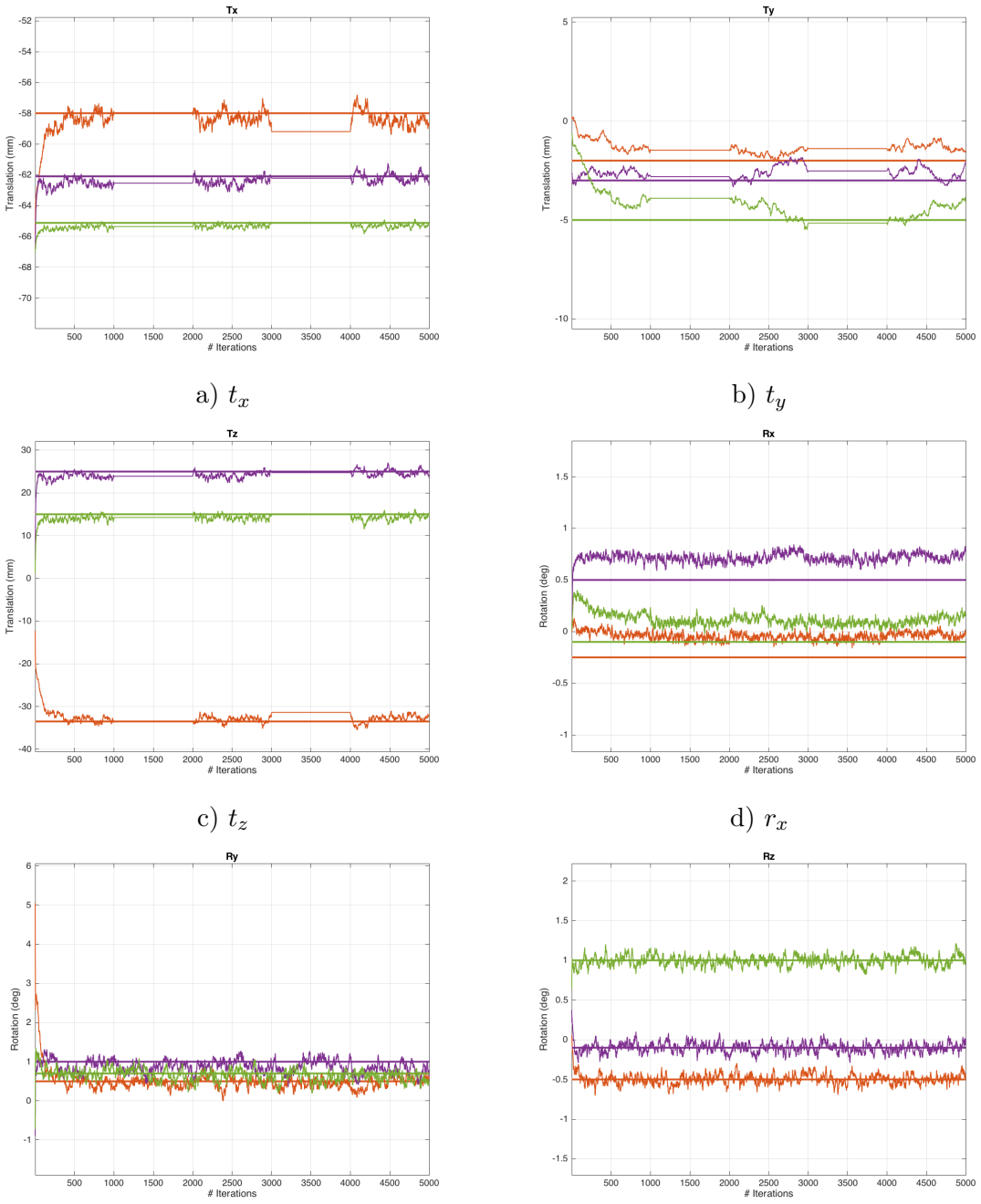
In this section we will compare the response of our multiple filter architecture and a classic filter architecture (single filter estimating all parameters at once). We started the analysis in a simulated environment with the cameras initialized at three different configurations (the eyes were looking to the right in experiment 1 and were looking to the

# Experiment	t_y(mm)	t_z(mm)	r_x(deg)	r_y(deg)	r_z(deg)
1 (ground-truth)	-2.00	-33.50	-0.25	0.50	-0.50
1 (mean multiple)	-1.44	-32.71	-0.05	0.46	-0.50
1 (mean classic)	-0.92	-30.24	-0.03	0.47	-0.49
1 (std multiple)	**0.17**	**0.99**	**0.03**	0.13	0.06
1 (std classic)	0.73	3.93	0.04	**0.13**	**0.05**
1 (mean error multiple)	**0.56**	**0.97**	**0.20**	0.11	0.05
1 (mean error classic)	1.08	3.30	0.22	**0.10**	**0.04**
2 (ground-truth)	-3.00	25.00	0.50	1.00	-0.10
2 (mean multiple)	-2.62	24.39	0.71	0.86	-0.10
2 (mean classic)	-2.21	19.22	0.72	0.97	-0.10
2 (std multiple)	**0.28**	**0.64**	**0.04**	0.16	**0.06**
2 (std classic)	0.48	6.10	0.04	**0.15**	0.06
2 (mean error multiple)	**0.40**	**0.73**	**0.21**	0.17	**0.05**
2 (mean error classic)	0.83	5.80	0.22	**0.12**	0.05
3 (ground-truth)	-5.00	15.00	-0.10	0.70	1.00
3 (mean multiple)	-4.53	14.40	0.10	0.63	1.00
3 (mean classic)	-4.44	11.12	0.12	0.63	1.00
3 (std multiple)	0.53	**0.57**	**0.04**	0.15	0.06
3 (std classic)	**0.33**	4.26	0.04	**0.14**	**0.05**
3 (mean error multiple)	**0.57**	**0.63**	**0.20**	0.13	0.05
3 (mean error classic)	0.58	3.94	0.22	**0.12**	**0.04**

Table 7.1: Estimates of the stereo parameters, in simulation, for comparison purposes, using selected observations for a multiple filter architecture and a classic filter architecture (3 experiments with 1 trial per experiment)

left in experiments 2 and 3). We then sampled virtual points from two different regions in space, region A and region B. Region A corresponds to points at a close distance, within a depth range $[500; 1500]$mm. Region B, corresponds to points far away, with a depth range $[10000; 20000]$mm. During the experiments, we were switching between these two regions with transitions happening at every 1000 iterations. The results are represented in figures 7.2 and 7.3 and table 7.1.

Points from region A (close range) are used from iteration $[0; 1000[$, $[2000; 3000[$ and $[4000; 5000[$ and points from region B (far range) are used from iteration $[1000; 2000[$ and $[3000; 4000[$. Points at far range can't be used to estimate the translation parameters as can be seen in figures 7.3a), 7.3b) and 7.3c). They induce drift oscillations in the translation parameters so it is preferable to not update the parameters estimates with these observations. Although points at infinity are not good for the translation parameters, they may be very informative for the rotational ones. Once again, only points

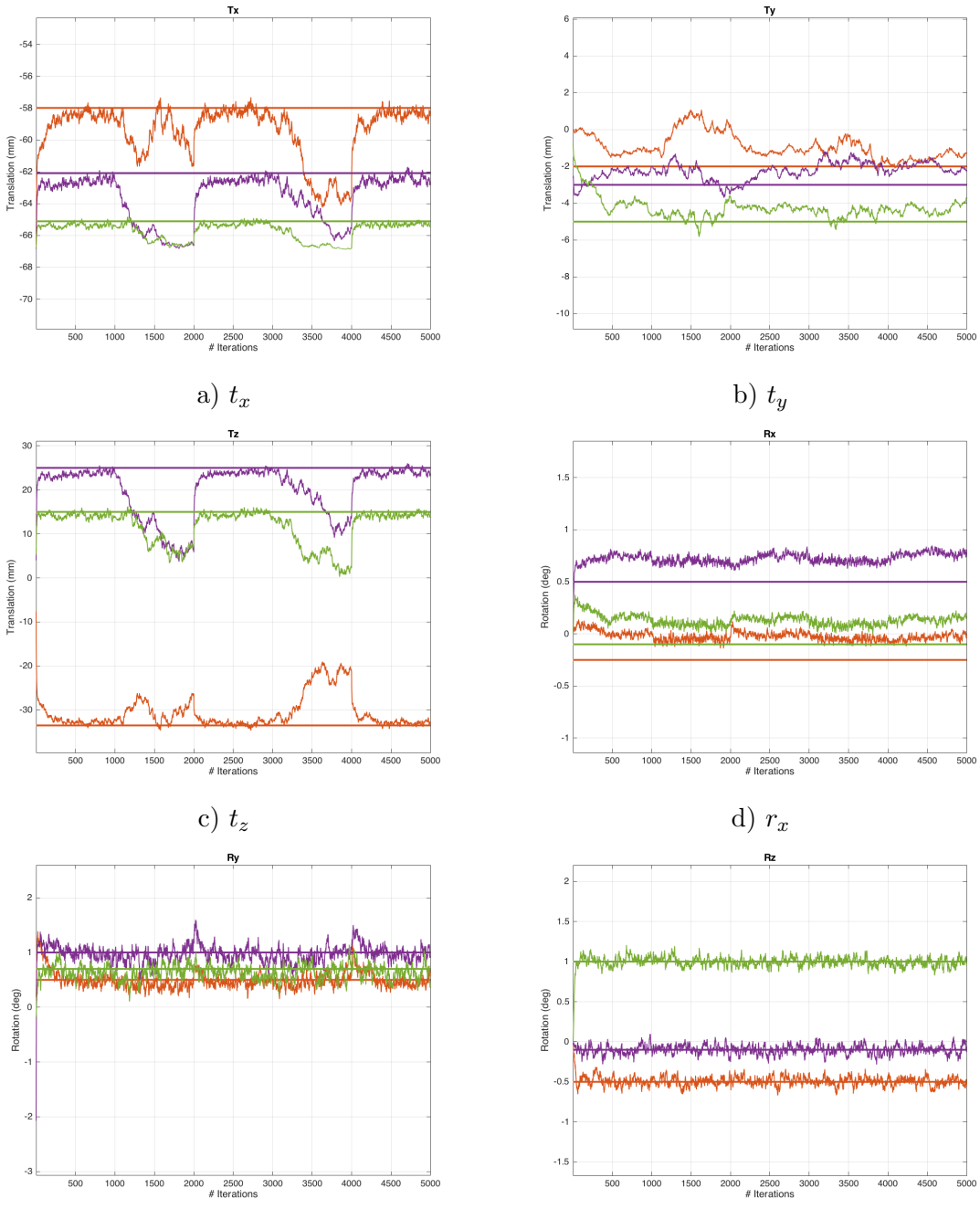


a) t_x b) t_y

c) t_z d) r_x

e) r_y f) r_z

Figure 7.2: Estimates of the stereo parameters, in simulation, for comparison purposes, using selected observations for a multiple filter architecture (3 experiments with 1 trial per experiment) - Experiment 1 (orange), Experiment 2 (green) and Experiment 3 (purple)



a) t_x b) t_y

c) t_z d) r_x

e) r_y f) r_z

Figure 7.3: Estimates of the stereo parameters, in simulation, for comparison purposes, using all observations for a classic filter architecture (3 experiments with 1 trial per experiment) - Experiment 1 (orange), Experiment 2 (green) and Experiment 3 (purple)

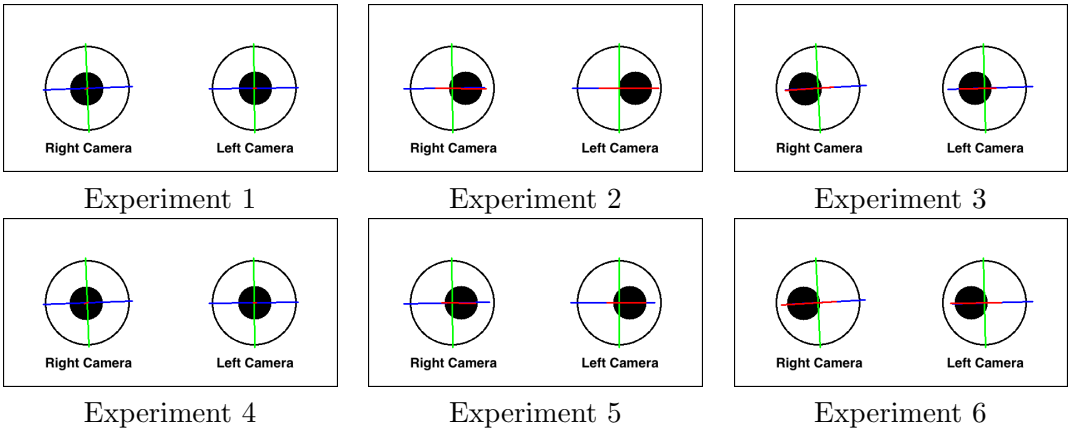


Figure 7.4: The configuration of the eyes for each experiment, to have a visual perception of the eyes position and orientation during stereo calibration.

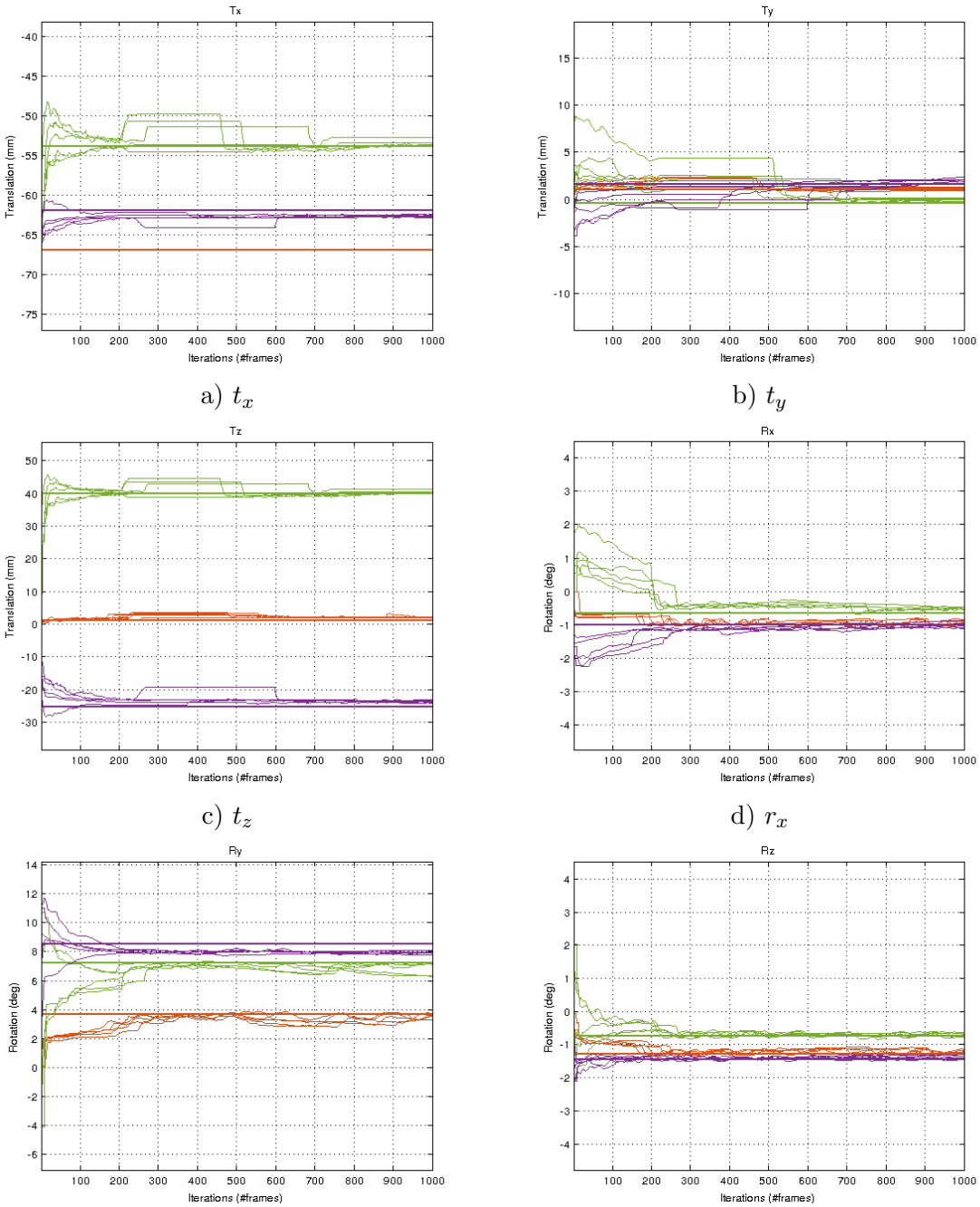
obeying to our observability model will be used, ensuring a more precise estimation of the parameters. It is clear the advantage of separating the system into multiple filters and feed each one with selected observations. The overall performance of the system becomes more stable, mainly for the translational parameters, with much lower errors (see table 7.1). The errors for the rotational parameters are very similar in both cases (multiple filters and single filter architectures) since we were only switching between close points and points at a far range, which mainly affect the translational parameters.

We performed the same analysis for the real stereo platform, where we initialized the cameras at three different configurations, as illustrated in figures 7.4a), 7.4b) and 7.4c) for experiments 1, 2 and 3 respectively.

We performed five trials for each experiment. The results are represented in figures 7.5 and 7.6 and table 7.2.

Once again, during the experiments we were switching between points from a close region and a far region, as seen in figure 7.7, although in this case the points at a far range are at a maximum depth of 3000mm.

A similar behaviour as the one found in simulation is present for both cases. When we use all observations, the estimate oscillates between a large range of values (for the translation parameters) when using points from region B (at far range). In some cases, it can be observed a coupling drift behaviour for rotation r_x and translation t_y since for some observations one can compensate the other, leading to a large variation in these two parameters. Even though we do not observe a clear improvement in precision for the rotational parameters, we ensure they keep their stability and converge more rapidly while avoiding estimation drift problems since ambiguities are always minimized.



a) t_x b) t_y

c) t_z d) r_x

e) r_y f) r_z

Figure 7.5: Estimates of the stereo parameters, with a real stereo platform, for comparison purposes, using selected observations for a multiple filter architecture (3 experiments with 5 trials per experiment) - Experiment 1 (orange), Experiment 2 (green) and Experiment 3 (purple)

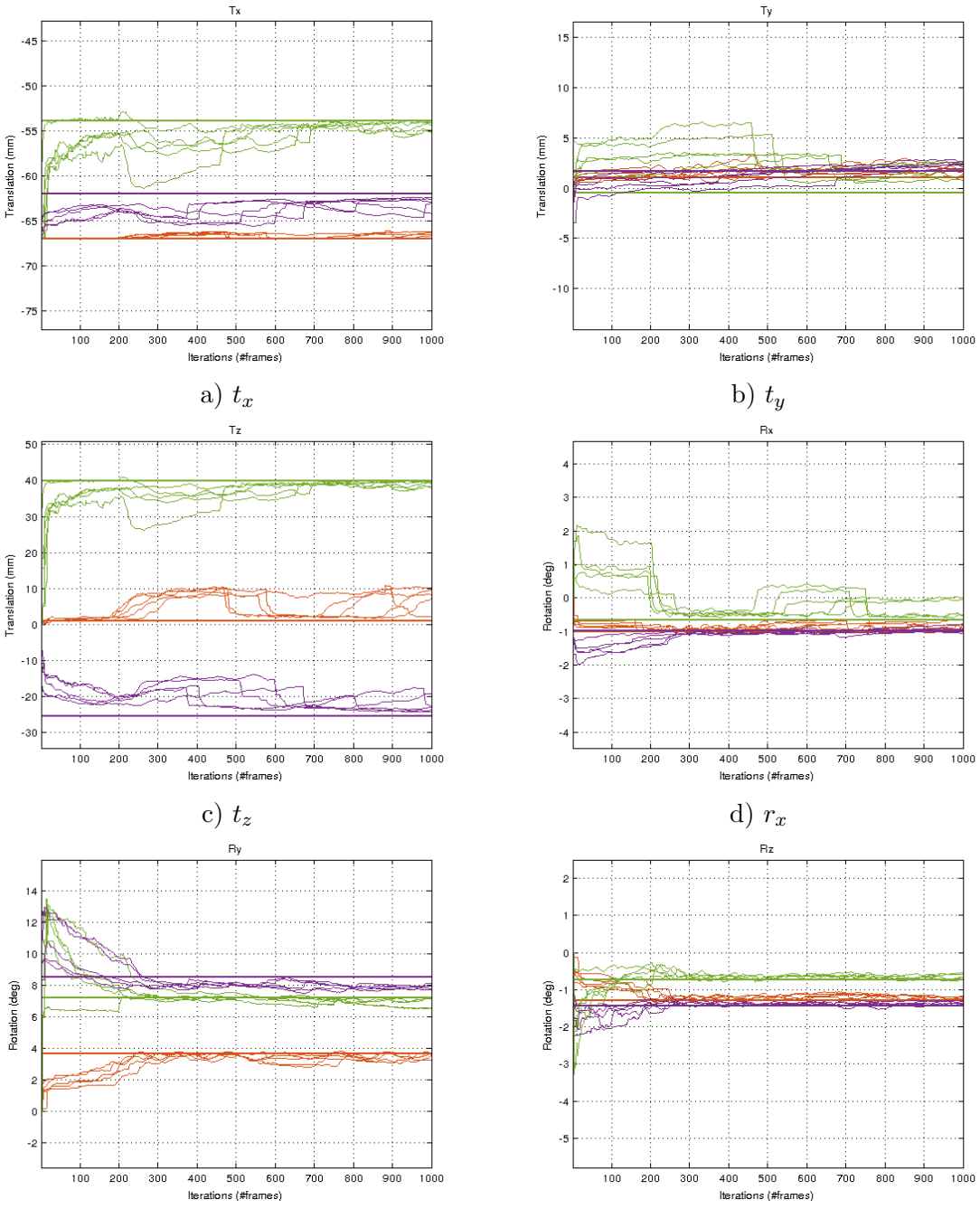
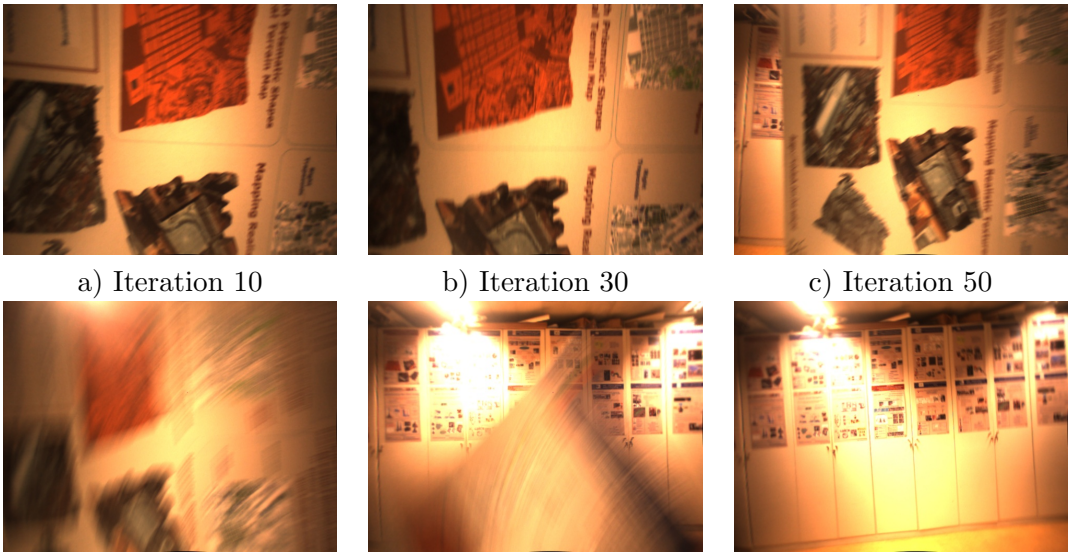


Figure 7.6: Estimates of the stereo parameters, with a real stereo platform, for comparison purposes, using all observations for a classic filter architecture (3 experiments with 5 trials per experiment) - Experiment 1 (orange), Experiment 2 (green) and Experiment 3 (purple)

# Experiment	t_y(mm)	t_z(mm)	r_x(deg)	r_y(deg)	r_z(deg)
1 (ground-truth)	1.05	0.95	-1.01	3.65	-1.29
1 (mean multiple)	1.15	2.06	-0.93	3.43	-1.23
1 (mean classic)	1.82	7.08	-0.85	3.45	-1.23
1 (std multiple)	**0.15**	**0.11**	**0.07**	**0.15**	0.04
1 (std classic)	0.39	2.87	0.10	0.18	**0.02**
1 (mean error multiple)	**0.10**	**1.10**	**0.07**	0.21	**0.05**
1 (mean error classic)	0.77	6.12	0.16	**0.19**	0.06
2 (ground-truth)	-0.46	39.80	-0.65	7.19	-0.74
2 (mean multiple)	-0.06	40.31	-0.50	6.82	-0.70
2 (mean classic)	1.74	38.86	-0.31	6.89	-0.66
2 (std multiple)	**0.16**	**0.52**	**0.03**	0.39	**0.05**
2 (std classic)	0.69	0.75	0.24	**0.28**	0.07
2 (mean error multiple)	**0.39**	**0.50**	**0.14**	0.36	**0.04**
2 (mean error classic)	2.20	0.94	0.33	**0.30**	0.08
3 (ground-truth)	1.62	-25.44	-0.99	8.50	-1.44
3 (mean multiple)	1.95	-23.67	-1.00	7.92	-1.38
3 (mean classic)	2.35	-22.18	-0.94	7.86	-1.35
3 (std multiple)	**0.21**	**0.38**	**0.08**	**0.10**	0.05
3 (std classic)	0.27	2.31	0.08	0.11	**0.04**
3 (mean error multiple)	**0.33**	**1.77**	**0.01**	**0.57**	**0.06**
3 (mean error classic)	0.73	3.26	0.04	0.63	0.08

Table 7.2: Estimates of the stereo parameters, with a real stereo platform, for comparison purposes, using selected observations for a multiple filter architecture and classic filter architecture (3 experiments with 5 trials per experiment).



a) Iteration 10 b) Iteration 30 c) Iteration 50

d) Iteration 60 e) Iteration 65 f) Iteration 70

Figure 7.7: Example of a stereo dataset acquisition, where we were switching between points at a close range, in this case the poster seen in figures a) to d), and points at a far range, in this case the closet seen in figures e) and f).

In these experiments we were switching between close points and points at far range, which explains the steps in the translational and rotational parameters estimates. The observations selection for the translational parameters uses the current estimates to calculate the real disparity between the image points. However, while presenting the system with close points, we decrease the region of overlap between the two images which results in a decreased number of points that are good for the rotational parameters. By not updating the rotational parameters, the translational ones can not be correctly updated as well. It was only after the rotational parameters converged that we were able to finally update the translational ones and converge to their correct values (thus reducing the steps size).

It is important to mention that when fed with good observations, both systems perform well. It is only when observations are not good that our solution can actuate by deciding not to use that information. By doing this we keep the estimates at their values and prevents drift.

# Experiment	t_y(mm)	t_z(mm)	r_x(deg)	r_y(deg)	r_z(deg)
1 (ground-truth)	-1.60	-3.75	-4.60	1.62	3.18
1 (mean)	-1.15	-4.16	-4.50	1.64	3.18
1 (std)	0.33	0.92	0.03	0.12	0.04
1 (mean error)	0.44	0.41	0.09	0.02	0.01
2 (ground-truth)	1.34	-21.10	-4.23	-0.83	4.28
2 (mean)	1.77	-20.67	-4.13	-0.81	4.27
2 (std)	0.27	0.70	0.03	0.10	0.04
2 (mean error)	0.43	0.42	0.09	0.01	0.01
3 (ground-truth)	0.06	17.62	3.25	2.95	0.02
3 (mean)	0.29	17.25	3.32	2.94	0.01
3 (std)	0.27	0.93	0.03	0.11	0.05
3 (mean error)	0.22	0.36	0.07	0.01	0.01
4 (ground-truth)	-0.40	2.68	0.94	0.52	4.19
4 (mean)	-0.13	2.61	1.02	0.51	4.17
4 (std)	0.47	0.66	0.03	0.11	0.04
4 (mean error)	0.26	0.06	0.08	0.01	0.01
5 (ground-truth)	-1.34	11.65	1.93	-0.25	-3.49
5 (mean)	-1.18	11.30	2.00	-0.25	-3.48
5 (std)	0.56	0.78	0.03	0.09	0.04
5 (mean error)	0.15	0.35	0.07	0.01	0.01

Table 7.3: Estimates of the stereo parameters, in simulation, for validation purposes, using selected observations for a multiple filter architecture (5 experiments with 5 trials per experiment).

7.2 Performance Characterization

To characterize our system's performance, we will assess its accuracy and repeatability. In simulation, we initialized the cameras at five different configurations, with the stereo parameters being randomly chosen from a Uniform Distribution, where t_y varied between -2mm and 2mm and t_z varied between -30mm and 30mm (t_x was directly obtained from the baseline constraint, with a baseline $B = 67$mm). We choose these variations to better simulate the real case where the cameras can only rotate around the y axis and any other rotation or translation are due to mounting errors of the sensors. The rotations were also sampled from a Uniform Distribution, with all the rotation values varying between -5° and 5°. For each experiment we performed five trials and the simulated points were obtained from a Uniform Distribution, within a region with a minimum depth of 250mm and a maximum depth of 3000mm. The results are represented in figure 7.8 and table 7.3.

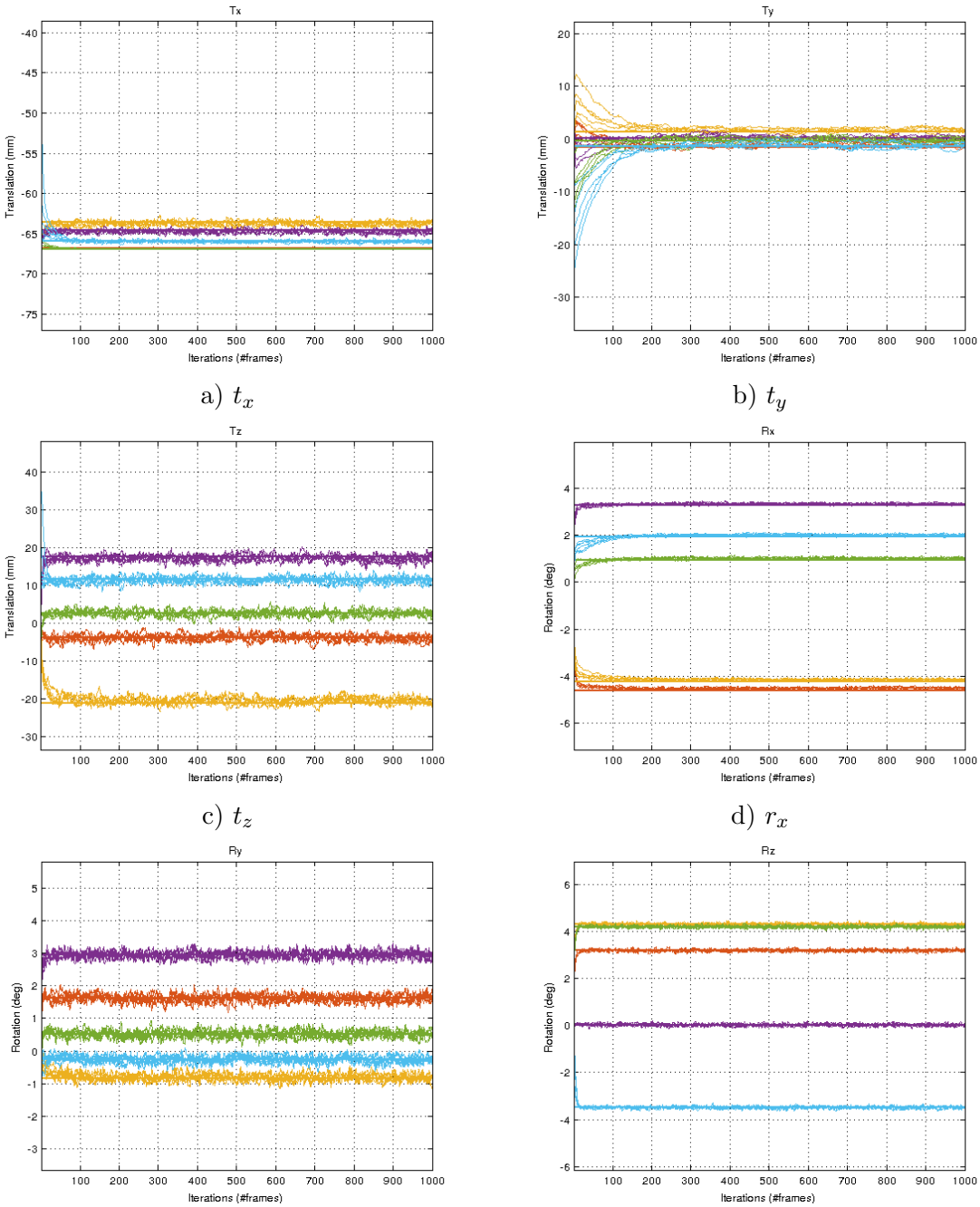


Figure 7.8: Estimates of the stereo parameters, in simulation, for validation purposes, using selected observations for a multiple filter architecture (5 experiments with 5 trials per experiment) - Experiment 1 (orange), Experiment 2 (yellow), Experiment 3 (purple), Experiment 4 (green) and Experiment 5 (cyan)

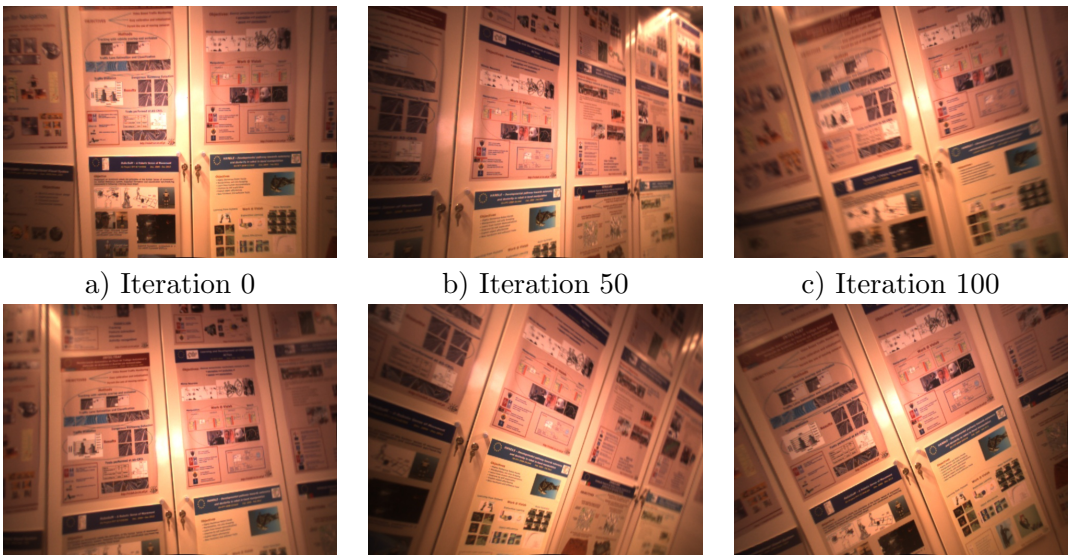
The system converges very rapidly to the correct parameters values, in less than 200 iterations for the slowest parameter t_y, or $\sim$ 13.3 seconds considering the frequency of the system with real observations ($\sim$ 15Hz). The error between the real and estimated parameters is very low showing larger variation for the rotational parameter r_y. This parameter is the one with the lowest observability considering its optimal region of acquisition is the smallest of the three rotational parameters, as seen in figure 6.5b), and points are rarely mapped to these regions, even in simulation. The lack of good points may lead to a slower convergence of the system for this particular parameter. In a real case scenario we must stimulate the system by acquiring different measurements in order to maximize the amount of useful information that will be used by the system. In our particular case we do this by rotating the robot's head making it look around and getting as much information from the surroundings as possible.

We performed five trials for each experiment, changing the image measurements but not the stereo parameters. As we can see from table 7.3, the maximum standard deviation we have, for the worst translational parameter, t_z, is 0.93mm with a mean absolute error of 0.37mm to the ground-truth value, or 2.1%. This demonstrates a good repeatability property of the system, showing its ability to converge within a 2% range to the same values under different operation conditions.

With the real stereo platform, we initialized the cameras at six different configurations, as seen in figure 7.4. For each experiment we performed five trials. Experiments 1 to 3 were already explained in section 7.1 and its results are represented in figure 7.5 and table 7.2. Experiments 4 to 6 were meant to test the system in a normal situation, having the cameras acquiring points from the environment without any constraint. Figure 7.9 shows an example of a dataset acquired for these experiments.

The results for these experiments are represented in figure 7.10 and table 7.4.

For experiments 1 to 3, the system presents very low estimation errors, with the largest being 1.77mm for the translational parameter t_z. In case of the rotational parameters, the largest mean error is present for r_y, with a value of 0.57deg. While in these experiments we can observe steps in the estimates caused by the switch between close points and points at a far range, for experiments 4 to 6 we can observe a very different response. For experiments 4 to 6, the system converges in less than 100 iterations (or $\sim$ 6s) and presents a more stable response (without any steps for the translational parameters) due to a better distribution of the features in the image within a depth range of $850 - 2500$mm. We have larger errors for the translational parameters, as expected, with the largest being for t_z, with a value of 3.50mm, still bellow the desired resolution of 5mm. The closest points are at a depth of 850mm which may not be enough for some



a) Iteration 0 b) Iteration 50 c) Iteration 100

d) Iteration 150 e) Iteration 200 f) Iteration 250

Figure 7.9: Example of stereo dataset acquisition without any constraint (here we can see the closet from the laboratory observed from different angles while rotating the head of the robot).

# Experiment	t_y(mm)	t_z(mm)	r_x(deg)	r_y(deg)	r_z(deg)
4 (ground-truth)	1.01	2.00	-0.84	3.96	-1.25
4 (mean)	3.14	4.75	-0.72	3.43	-1.20
4 (std)	0.35	0.48	0.02	0.14	0.04
4 (mean error)	2.12	2.75	0.11	0.53	0.04
5 (ground-truth)	0.01	25.34	-0.66	3.39	-0.98
5 (mean)	2.49	28.84	-0.54	2.93	-0.95
5 (std)	0.40	1.28	0.04	0.13	0.03
5 (mean error)	2.48	3.50	0.12	0.45	0.02
6 (ground-truth)	1.70	-33.53	-0.82	3.50	-1.69
6 (mean)	4.57	-34.70	-0.81	3.00	-1.62
6 (std)	0.25	1.08	0.03	0.10	0.05
6 (mean error)	2.86	1.17	0.01	0.50	0.06

Table 7.4: Estimates of the stereo parameters, with the real platform, for validation purposes, using selected observations for a multiple filter architecture (3 experiments with 5 trials per experiment).

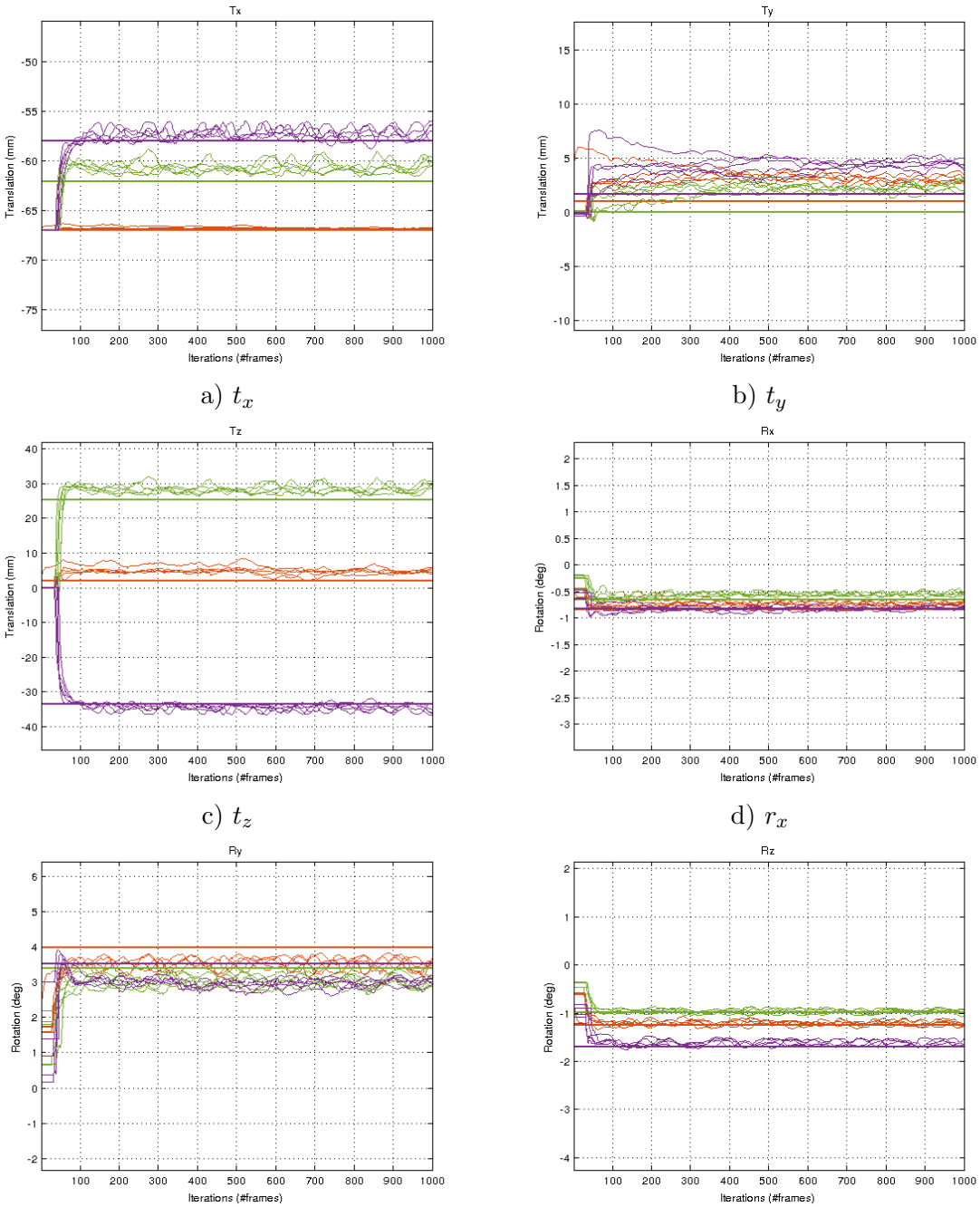


Figure 7.10: Estimates of the stereo parameters, with the real platform, for validation purposes, using selected observations for a multiple filter architecture (3 experiments with 5 trials per experiment) - Experiment 4 (orange), Experiment 5 (green) and Experiment 6 (purple)

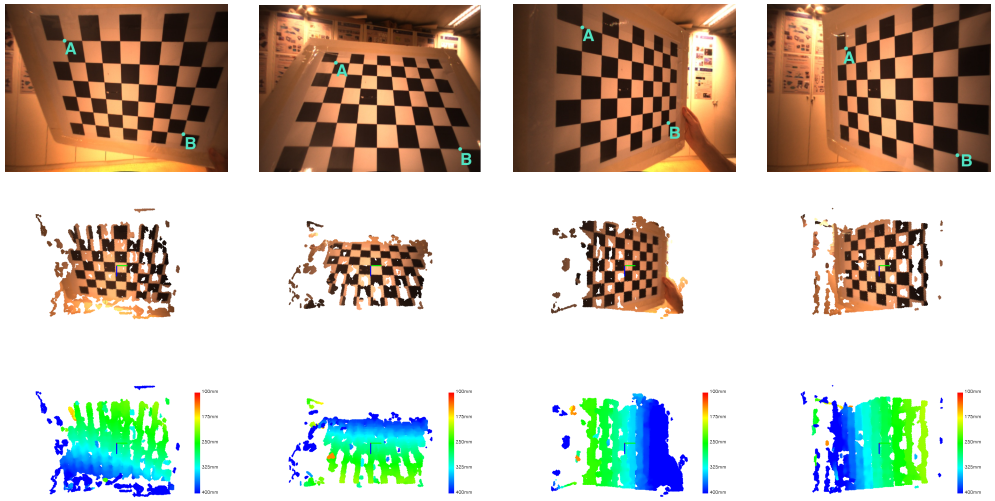


Figure 7.11: Stereo calibration estimates using the classical filter architecture on normal measurements (5 trials for each experiment)

values of t_y and t_z. The rotational parameters present similar errors as in the previous three experiments, with the largest being, again, for rotational parameter r_y. Due to observations selection, for the six experiments the system was only using around 50% of the total measurements for the translational parameters and 30% for the rotational ones, in average. This shows the efficiency of the system in using less observations.

In summary, both for simulated and real experiments, we performed five trials to test the repeatability of the system, without any change of the stereo parameters. The system always converged to similar values as shown by the low standard deviations in tables 7.2 and 7.4.

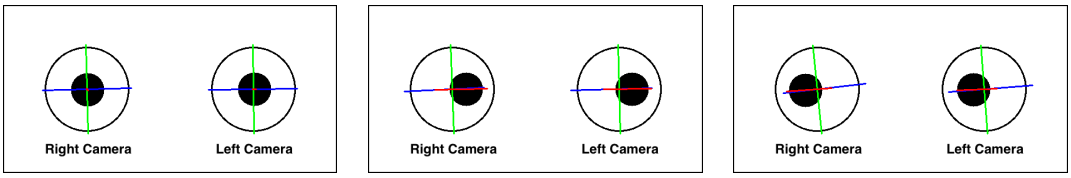
7.3 3D Reconstruction

To complete the accuracy analysis of our stereo calibration system, we reconstructed several scenes from different groups of stereo image pairs, acquired for multiple configurations of the cameras. The first analysis was performed for experiments 1 to 3, whose stereo roto-translation is represented in table 7.2. For this analysis we used a calibration pattern where we set a fixed length $AB = 361.3$mm, as seen in figure 7.11. The reconstructed length AB for the different experiments is represented in table 7.5

The reconstructed points were obtained, for the 4 orientations of the pattern, using the method described in section 3.4. The reconstruction results are very accurate with a

# Experiment	Mean (mm)	Std (mm)	Mean Error (mm)
1	359.90	1.69	1.4
2	356.55	1.74	4.75
3	363.56	4.15	2.26

Table 7.5: Reconstructed length AB for experiments 1, 2 and 3, considering all the five trials.



Experiment 7 Experiment 8 Experiment 9

Figure 7.12: The configuration of the eyes for each experiment, to have a visual perception of the eyes position and orientation during stereo reconstruction.

maximum mean error of 4.75mm for a target whose depth is changing between 200mm and 400mm.

The second analysis consisted in a full scene reconstruction with several objects at different depths, as seen in figure 7.13. We reconstructed the vector norm from the left camera optical center O to points A, B, C and D for three configurations of the cameras, as seen in figure 7.12 with the calibrated roto-translation values represented in table 7.6. The reconstructed vector norms for each point and different cameras configuration are represented in table 7.7.

The accuracy of the stereo reconstruction is clearly visible for different configurations of the cameras and for points at different depths, with a maximum reconstruction error of 5.12mm for a point at 475mm. Once again, we used the triangulation method described in section 3.4 which may cause additional errors caused by the manual selection of the points. These experiments allows us to show the capability of our system to calibrate any

Figure 7.13: Stereo reconstruction of a full scene with different objects at different depths

# Experiment	t_x (mm)	t_y (mm)	t_z (mm)	r_x (deg)	r_y (deg)	r_z (deg)
7	-66.95	0.773	2.39	-0.72	1.33	0.99
8	-59.06	1.667	31.59	-0.074	-2.81	-0.95
9	-61.559	5.21	-25.93	-0.46	3.33	-1.47

Table 7.6: The calibrated parameters, given by the system, for experiments 7, 8 and 9.

# Experiment	$\|OA\|$	$\|OB\|$	$\|OC\|$	$\|OD\|$
7 (ground-truth)	337mm	340mm	385mm	490mm
7 (calibrated)	338.80mm	341.77mm	384.99mm	486.85mm
7 (mean error)	1.80mm	1.77mm	0.01mm	3.15mm
8 (ground-truth)	310mm	330mm	380mm	475mm
8 (calibrated)	310.05mm	327.53mm	379.97mm	469.88mm
8 (mean error)	0.05mm	2.47mm	0.03mm	5.12mm
9 (ground-truth)	365mm	350mm	395mm	510mm
9 (calibrated)	364.99mm	349.97mm	394.13mm	509.97mm
9 (mean error)	0.01mm	0.03mm	0.87mm	0.03mm

Table 7.7: Comparison of reconstructed depths for the different experiments.

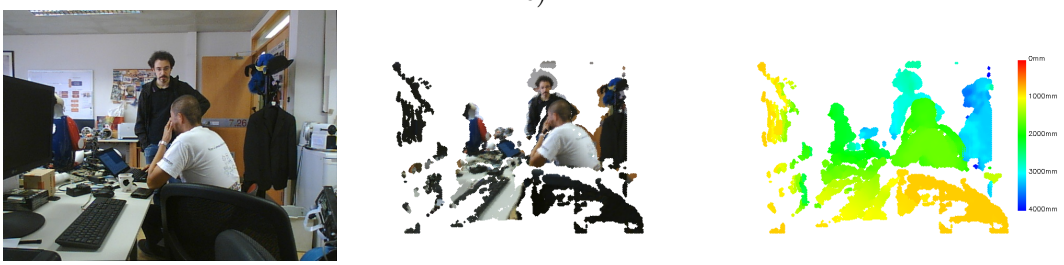
stereo platform equipped with two cameras, regardless of their position and orientation. Such a system is of utmost importance for the computer vision community and the robotics world.

In figure 7.14 we present more examples of stereo reconstructions obtained for different configurations of the cameras, with an accurate reconstruction of the scene with all the elements at their correct depths. These other examples show the robustness of our system which is able to correctly calibrate any stereo platform regardless of the cameras nature or orientation.

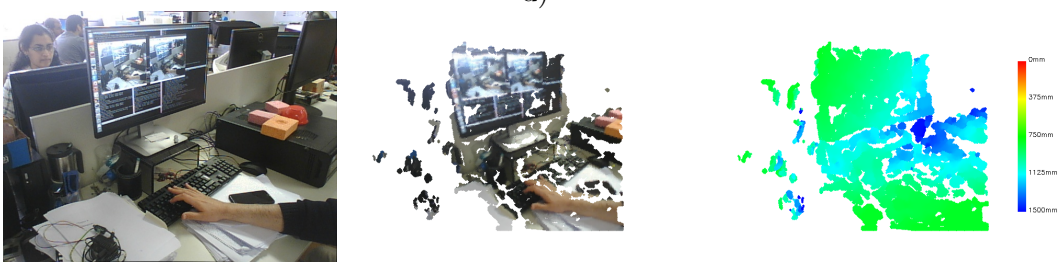
a)

b)

c)



d)



e)

Figure 7.14: Stereo reconstruction examples for different configurations of the cameras.

7.4 Calibrated Internal Model

In this final section we will show how the previously presented head and stereo calibration systems could fully calibrate the internal model of a robot head, by showing a practical and real application. To fully understand the surroundings a robotic platform requires a calibrated internal model and a full representation of the environment. These two models are highly dependent on each other where a good world representation is only possible if the robotic platform can position itself in the world and know its state at every time instant. Considering the internal model calibration problem is already solved, we will address in this section the world representation in a robot centered reference frame, the egosphere.

An egosphere consists of a tri-dimensional representation of world points within a common reference frame, in our case the robot's neck base. By rotating the head we can build this 3D map by just adding new points to the egosphere. The head calibration system will ensure the points location are consistent with the absolute zero position of the robot's head independently of its orientation. The stereo calibration system will ensure the 3D position of each reconstructed point is consistent with its real world position relative to the reference frame's origin. The calibration of both systems allows the construction of an accurate egosphere that is extremely useful in many applications.

In [10, 35] the authors present an egosphere where the world's 3D points are projected into a geodesic spherical surface. The authors then show the advantages of having such a representation mainly in attention tasks where the robot needs to gaze certain objects represented in the egosphere with very low latency. Their robotic platform is much more simpler with a pan-tilt structure and a single camera as end-effector. In our thesis we have constructed an egosphere from raw data where the 3D world points were directly mapped into the base reference frame using solely calibrated kinematic information and stereo information. The dense disparity maps required for the 3D reconstruction were obtained using the semi-global block matching algorithm, presented in [21]. Figures 7.15 and 7.16 show two examples of our 3D egosphere representations, using our 3DPointCloud visualization tool, where the robotic head was initialized in two different configurations, for experiments 10 and 11. We acquired 1000 image frames for each experiment while rotating the robot's head. The calibration results are represented in tables 7.8 and 7.9.

It becomes very clear the advantage of having a representation of this kind with this level of precision from raw data (without any posterior alignment of the point cloud). The importance of a fully calibrated internal model is tremendous and ensures the platform

# Experiment	δ_0(deg)	δ_1(deg)	δ_2(deg)	δ_3(deg)	δ_4(deg)	δ_5(deg)
10	-40.71	-38.03	40.74	2.38	44.60	-47.60
11	-51.44	33.10	56.86	-5.74	-47.58	-40.87

Table 7.8: Head calibration results for both experiments.

# Experiment	t_x(mm)	t_y(mm)	t_z(mm)	r_x(deg)	r_y(deg)	r_z(deg)
10	-66.53	-0.17	7.89	-1.23	0.39	-1.16
11	-66.39	-0.53	9.00	-1.27	0.87	-1.26

Table 7.9: Stereo calibration results for both experiments.

Figure 7.15: Experiment 10 - the full reconstruction can be seen in this video: `https://www.youtube.com/watch?v=2C7cUxvsFzo`

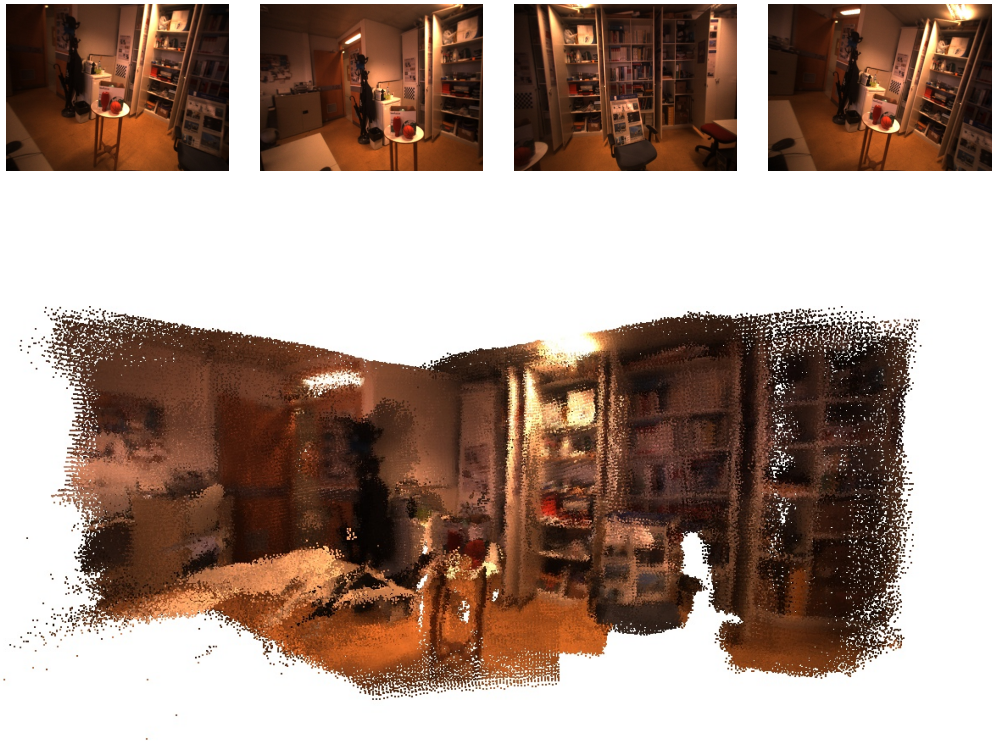


Figure 7.16: Experiment 11 - the full reconstruction can be seen in this video: `https://www.youtube.com/watch?v=hSqrj4ENyJk`

will be prepared to accomplish the tasks it was designed for a wide range of operational conditions.

Chapter 8

Conclusions

This thesis focus on the complete calibration of the internal model of a robot's head using only information from embedded sensors and non-linear filtering techniques. We concentrated our work in two main areas: the kinematic calibration of the robot head joints and the stereo calibration of the cameras.

Our first contribution is a calibration system at a kinematic level to be applied when the joints are equipped with relative encoders. The proposed system is able to rapidly estimate the offsets for each joint by using a non-linear filter together with information from the encoders, the IMU and the cameras. The sensor fusion allows the correct estimation of the joint offsets for different operational conditions, making the system robust and extremely adaptable. The results show an accurate calibration system that can easily calibrate any kinematic platform within a few iterations, which is important when the robotic platform is used on a daily basis, requiring a calibration procedure every time.

Our second contribution is a stereo calibration system that can correctly calibrate any stereo system in a very rapid and accurate way without using any markers or special patterns. The proposed system uses natural information from the environment to achieve a calibration status in a few iterations, being almost $40\times$ faster to achieve calibration than the Bouguet Toolbox used as ground-truth in this work, with similar accuracy (considering it took us 10min per experiment, in average, to calibrate the stereo system using this toolbox). An important feature of our system is the separation in multiple sub-filters that estimate each parameter of the stereo roto-translation in separate. This separation was proposed due to the observability analysis performed in our work that showed that each parameter under estimation requires different types of measurements. The translation parameters require points that are close to the cameras while the rota-

tional parameters require points located at specific regions of the image. This analysis allowed us to select which measurements were better for each parameter in separate and feed each filter separately. The results show that measurements selection in multiple-filter architecture is much better than trying to estimate all the parameters at once using all the acquired measurements, as performed in the classical approach. The analysis is extremely useful for those who want to better understand high accuracy stereo calibration since it shows quality is better than quantity. Most stereo calibration systems tend to use as many measurements as possible which may be harmful for the final estimates, as already shown in our work. Some measurements may deteriorate the estimates since they are providing little or no information about the real state of the system and are only working as noise.

Finally we have shown how these two systems could be integrated to fully calibrate the internal model of a robot's head, in our case the iCub, from the robot's neck to the eyes. The calibrated internal model always knows its end-effector (the cameras) position and orientation allowing the creation of an accurate egosphere centered at the base of the kinematic chain (in the case of the iCub head it is located at the base of the neck). A representation of this kind with the presented level of precision from raw data (without any posterior alignment of the point cloud) is of utmost importance for many applications and tasks requiring a 3D representation of the environment in a robot centered reference frame. In an assembly line where robotic platforms (usually arms) and humans share the same space during operation, it is important that a robotic arm can detect a person in its working area for safety reasons. A full representation of its working area will provide all the information the robot needs to detect a person and avoid any accident that may occur. These robotic arms require well calibrated internal models in order to predict expected events and detect unexpected ones while performing the tasks they were designed for. The solution presented in [30] based on the EP concept would generate the desired output for a safer cooperation between robots and humans within the same environment. With a full representation of the surroundings under the assumption of a static environment, we can predict visual information using the robot's movements and rapidly compare the predicted and real images with the intent of detecting unpredictable events.

The egosphere representation jointly with a calibrated internal model and an accurate stereo vision would be very useful also in grasping and manipulation tasks where the robot must interact with elements from a closed environment. These tasks are of an extremely complex execution where an accurate calibration, for the kinematic chain and the stereo vision, would help the correct accomplishment of the tasks, using a solution

such as the one presented in [43]. Our calibration systems would provide the correct head orientation and accurate stereo information to the eye-hand model described in the mention work. These are only a few examples of how important a calibrated internal model is for most robotic tasks, from manipulation, to grasping or just for safety reasons in human-robot interaction and show the magnitude of such a system.

8.1 Future Work

The selection of good points to calibrate the platform is a helpful method to prevent drift on the parameters but works in a passive manner, i.e. if there are no good points during a significant period of time, the parameters get "frozen" and do not adapt to eventual changes. For example, if is a robot walking forward in open space having the horizon in the center of the image, it has no information to adapt its translational and some of the rotational parameters. Therefore, an active strategy must be developed to execute purposive movements of the robot so that visual features are gathered at good configurations, thus taking advantage of active vision principles. In the given example, looking at the floor or at own robot body parts will provide nearby cues to update the translation parameters. Also, controlling the head-eye system to have horizon points higher or lower in the view field will help the estimation of the rotational parameters.

Another direction worth exploring is to use lines rather than points as measurements for the calibration. The detection and matching of lines is more robust to outliers that points, thus potentially improving the accuracy and convergence of the method.

Appendices

Appendix A

Observability Analysis: Closed Form Solution

For this observability analysis, we will always start the cameras from the nominal position and will apply a small δ to each stereo parameter alone, to see which points generate the largest vertical distance to the nominal epipolar lines. Let's consider the roto-translation ${}^{R}T_{L}$ between the left and right images, represented in (3.3). For this analysis we will consider both cameras have optical centers given by c_x and c_y and focal lengths represented as f_x and f_y. The left world point used has generic coordinates $\left[X^L, Y^L, Z^L\right]$. To map a world point $\left[X^L, Y^L, Z^L\right]$ seen in the left camera reference frame to an image point $\left[u^R, v^R\right]$ in the right image we must first apply the roto-translation ${}^{R}T_{L}$:

$$\left[X^R, Y^R, Z^R, 1\right]^T = {}^{R}T_{L} . \left[X^L, Y^L, Z^L, 1\right]^T \tag{A.1}$$

The right image point $\left[u^R, v^R\right]$ can be obtained from the projective model applied to the transformed world point $\left[X^R, Y^R, Z^R\right]$:

$$\left[\ u^R, v^R, 1\ \right]^T = K \left[\ x^R, y^R, 1\ \right]^T \tag{A.2}$$

where x^R and y^R correspond to the normalized coordinates of the point, given by:

$$\left[x^R, y^R\right] = \left[X^R/Z^R, Y^R/Z^R\right] \tag{A.3}$$

and K corresponds to the intrinsic matrix, [17]. We can define the vertical coordinate v^R as a function of all the stereo parameters, t_y, t_z, r_x, r_y and r_z and a generic left point in the world, $\left[X^L, Y^L, Z^L\right]$:

$$v^R = f\left(t_y, t_z, r_x, r_y, r_z, X^L, Y^L, Z^L\right) \tag{A.4}$$

A.1 Observability of translational parameter t_y

Let's consider the influence of the translational parameter t_y in the vertical image coordinate v^R on the right camera for a generic world point, following the equations (A.1) and (A.2), when t_z, r_x, r_y and r_z are equal to 0. The vertical coordinate v_r becomes:

$$v^R\left(t_y, Y^L, Z^L\right) = c_y + \frac{f_y\left(Y^L + t_y\right)}{Z^L} \tag{A.5}$$

where we are omitting from the equation all the parameters that were set to zero. The vertical displacement d_v when we apply a variation δ in t_y from its nominal position is given by:

$$d_v = \left|v^R\left(\delta, Y^L, Z^L\right) - v^R\left(0, Y^L, Z^L\right)\right| = \left|\frac{f_y\delta}{Z^L}\right| \tag{A.6}$$

We are interested in points where $d_v > \mathcal{E}$ with $\mathcal{E} \in \mathbb{N}_{>0}$ representing the variation in pixel units. Considering the case where $\mathcal{E} = 1$ corresponding to the minimum observable variation given the pixel discretization, we can see that points at distances Z^L larger than $|f_y\delta|$ will not generate any variation in the image coordinate v^R.

A.2 Observability of translational parameter t_z

The same analysis can be done for the parameter t_z. The vertical coordinate v^R when t_y, r_x, r_y and r_z are equal to 0 is given by:

$$v^R\left(t_z, Y^L, Z^L\right) = c_y + \frac{f_y Y^L}{Z^L + t_z} \tag{A.7}$$

By applying a variation δ in t_z the vertical displacement d_v is given by:

$$d_v = \left|v^R\left(\delta, Y^L, Z^L\right) - v^R\left(0, Y^L, Z^L\right)\right| = \left|\frac{f_y\left(Y^L\delta\right)}{\left(Z^L\right)^2 + \delta z^L}\right| \tag{A.8}$$

This equation can be simplified by dividing both terms by Z^L and applying the projective model of the camera, where $v'^L = f_y.Y^L/Z^L = v^L - c_y$. Thus the final equation for the vertical displacement d_v is given by:

$$d_v = \left| \frac{{v'}^L \delta}{Z^L + \delta} \right| \tag{A.9}$$

This way we can easily relate the vertical displacement d_v with the image location of the point and its depth, simplifying the analysis. Assuming the points' depth Z^L is greater than δ for most operation conditions, we can simplify the equation, which becomes:

$$d_v = \frac{\left| {v'}^L \delta \right|}{Z^L + \delta} \tag{A.10}$$

Just like in the previous case, the boundary conditions for an optimal operation are those where $d_v > \mathcal{E}$ with $\mathcal{E} \in \mathbb{N}_{>0}$ representing the variation in pixel units. This condition gives an expression for the maximum point's depth Z^L for any point ${v'}^L$:

$$Z^L < \frac{\left| \delta {v'}^L \right|}{\mathcal{E}} - \delta \tag{A.11}$$

We can also see that points near the bottom or top of the image allow for larger depths (given the ${v'}^L$ term), where the vertical displacement d_v reaches it's maximum.

A.3 Observability of rotational parameter r_x

The influence of the rotational parameter r_x in the image point coordinate v^R when t_y, t_z, r_y and r_z are equal to 0 is shown in the following equation:

$$v^R \left(r_x, Y^L, Z^L \right) = c_y + \frac{f_y \left(Y^L \cos\left(r_x\right) - Z^L \sin\left(r_x\right) \right)}{Z^L \cos\left(r_x\right) + Y^L \sin\left(r_x\right)} \tag{A.12}$$

where for a single rotation r_x, the rotation matrix ${}^R R_L$, given by the Rodrigues Rotation Formula [37], is represented as:

$${}^R R_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(r_x\right) & -\sin\left(r_x\right) \\ 0 & -\sin\left(r_x\right) & \cos\left(r_x\right) \end{bmatrix} \tag{A.13}$$

By applying a small rotation δ to r_x, the variation in the vertical image coordinate is given by:

$$d_v = \left|v^R\left(\delta, Y^L, Z^L\right) - v^R\left(0, Y^L, Z^L\right)\right| = \left|\frac{f_y\left(\left(Z^L\right)^2\sin\left(\delta\right) + \left(Y^L\right)^2\sin\left(\delta\right)\right)}{\left(Z^L\right)^2\cos\left(\delta\right) + Y^L Z^L\sin\left(\delta\right)}\right| \quad \text{(A.14)}$$

If we divide the two terms of equation (A.14) by $\left(Z^L\right)^2$ we have:

$$d_v = \left|\frac{f_y\left(\sin\left(\delta\right) + \left(y^L\right)^2\sin\left(\delta\right)\right)}{\cos\left(\delta\right) + y^L\sin\left(\delta\right)}\right| \quad \text{(A.15)}$$

where $y^L = Y^L/Z^L$ corresponds to the normalized image point projected in the left image plane. This equation is of the form $\left|\frac{K_1}{K_2}\right|$ where, in order for the condition $d_v > \mathcal{E}$ to be fulfilled, one of these two cases must occur: i) $\frac{K_1}{K_2} > \mathcal{E}$ if $\delta > 0$ or ii) $\frac{K_1}{K_2} < -\mathcal{E}$ if $\delta < 0$.

In case i), the equation for y^L is given by:

$$\left(y^L\right)^2 f_y\sin\left(\delta\right) - \mathcal{E}y^L\sin\left(\delta\right) + f_y\sin\left(\delta\right) - \mathcal{E}\cos\left(\delta\right) > 0 \quad \text{(A.16)}$$

This second order equation will provide two conditions for the variable y^L, given by:

$$y^L > \frac{\mathcal{E}\sin\left(\delta\right) \pm \sqrt{\lambda}}{2f_y\sin\left(\delta\right)} \quad \text{(A.17)}$$

with λ represented as:

$$\lambda = \mathcal{E}^2\sin\left(\delta\right)^2 - 4f_y^2\sin\left(\delta\right)^2 + 4\mathcal{E}f_y\sin\left(\delta\right)\cos\left(\delta\right) \quad \text{(A.18)}$$

which gives an interval of restrictions for δ (thus for r_x) from the fact that $\lambda > 0$:

$$0 < \delta < \tan\left(\frac{4\mathcal{E}f_y}{4f_y^2 - \mathcal{E}^2}\right)^{-1} \quad \text{(A.19)}$$

For values of δ greater than $\tan\left(\frac{4\mathcal{E}f_y}{4f_y^2-\mathcal{E}^2}\right)^{-1}$ the variation on d_v is always larger than $\mathcal{E}$pixel making the condition always true. However, for values of δ within the interval, we can see from the condition in (A.17) that y^L has two boundaries setting the minimum and maximum image coordinates that can generate a variation in d_v larger than $\mathcal{E}$. The greater the value of y^L the larger the variation (it grows quadratically with y^L) meaning image points closer to the top and bottom borders of the left image will generate larger variations on the right image for any rotation along r_x. The image coordinates can easily

be obtained by applying the intrinsics to y^L. The same analysis can be done for case ii) where the equation for y^L is given by:

$$y^L < \frac{-\mathcal{E}\sin(\delta) \pm \sqrt{\lambda}}{2f_y \sin(\delta)} \tag{A.20}$$

with λ represented as:

$$\lambda = \mathcal{E}^2 \sin(\delta)^2 - 4f_y^2 \sin(\delta)^2 - 4\mathcal{E} f_y \sin(\delta)\cos(\delta) \tag{A.21}$$

where the interval of restrictions for δ (thus for r_x), in this case is given by:

$$\tan\left(-\frac{4\mathcal{E} f_y}{4f_y^2 - \mathcal{E}^2}\right)^{-1} < \delta < 0 \tag{A.22}$$

Just like the previous case, for values of δ lower than $\tan\left(-\frac{4\mathcal{E}f_y}{4f_y^2-\mathcal{E}^2}\right)^{-1}$ the variation on d_v is always larger than $\mathcal{E}$pixel. For values of δ within the interval we have two boundaries setting the minimum and maximum image coordinates that can generate a variation in d_v larger than $\mathcal{E}$. These conditions give us the boundaries for the optimal regions where the points to estimate r_x must be acquired, considering the desired SNR.

A.4 Observability of rotational parameter r_y

The influence of the rotational parameter r_y in the image coordinate v^R when t_y, t_z, r_x and r_z are equal to 0 is represented by the following equations:

$$v^R\left(r_y, X^L, Y^L, Z^L\right) = c_y + \frac{f_y Y^L}{Z^L\cos(r_y) + X^L\sin(r_y)} \tag{A.23}$$

where for a single rotation r_y, the rotation matrix ${}^R R_L$ is represented as:

$${}^R R_L = \begin{bmatrix} \cos(r_y) & 0 & -\sin(r_y) \\ 0 & 1 & 0 \\ \sin(r_y) & 0 & \cos(r_y) \end{bmatrix} \tag{A.24}$$

The same variation previously explained can be applied to these new coordinates and is given by:

$$d_v = \left|v^R\left(\delta, X^L, Y^L, Z^L\right) - v^R\left(0, X^L, Y^L, Z^L\right)\right| = \left|\frac{f_y\left(Y^L Z^L \cos\left(\delta\right) + Y^L X^L \sin\left(\delta\right) - Z^L Y^L\right)}{\left(Z^L\right)^2 \cos\left(\delta\right) + X^L Z^L \sin\left(\delta\right)}\right| \tag{A.25}$$

We can divide each term of this equation by $\left(Z^L\right)^2$, ending up with:

$$d_v = \left|\frac{f_y\left(y^L \cos\left(\delta\right) + x^L y^L \sin\left(\delta\right) - y^L\right)}{\cos\left(\delta\right) + x^L \sin\left(\delta\right)}\right| \tag{A.26}$$

where x^L and y^L correspond to the normalized image point coordinates projected in the left image plane.

This equation is of the form $\left|\frac{K_1}{K_2}\right|$ where, in order for the condition $d_v > \mathcal{E}$ to be fulfilled, one of these two cases must occur: i) $\frac{K_1}{K_2} > \mathcal{E}$ or ii) $\frac{K_1}{K_2} < -\mathcal{E}$.

In case i) and observing the previous equation we have the following condition for x^L:

$$x^L < \frac{f_y y^L \cos\left(\delta\right) - f_y y^L - \mathcal{E}\cos\left(\delta\right)}{\mathcal{E}\sin\left(\delta\right) - f_y y^L \sin\left(\delta\right)} \tag{A.27}$$

This condition creates the first set of boundaries for x^L that are influenced by the sign of δ. If $\delta > 0$ for r_y, this condition generates two regions close to the top left and bottom right corners of the image, depending if the coordinate y^L is negative or positive, respectively. If $\delta < 0$, the condition generates two regions close to the top right and bottom left corners of the image, again depending if y^L is negative or positive, respectively.

For case ii), the equation is given by:

$$x^L > \frac{f_y y^L - f_y y^L \cos\left(\delta\right) - \mathcal{E}\cos\left(\delta\right)}{\mathcal{E}\sin\left(\delta\right) + f_y y^L \sin\left(\delta\right)} \tag{A.28}$$

This other equation creates another set of boundaries for x^L that are influenced, again, by the sign of δ. If $\delta > 0$ for r_y, this condition generates two regions close to the top right and bottom left corners of the image, depending if y^L is negative or positive, respectively. If $\delta < 0$, the condition generates two regions close to the top left and bottom right corners of the image, again depending if y^L is negative or positive, respectively. These two conditions will give us information about the quality of a point by observing if it falls into the regions delimited by the boundaries. The largest variations occur for points near the four corners of the image where $\left|x^L\right|$ and $\left|y^L\right|$ have higher values. Points

that do not obey to these condition will not generate any variation in d_v.

A.5 Observability of rotational parameter r_z

Let's consider the vertical image coordinate of a generic world point when a rotation r_z is applied and t_y, t_z, r_x and r_y are equal to 0:

$$v^R\left(r_z, X^L, Y^L, Z^L\right) = c_y + \frac{f_y\left(Y^L\cos\left(r_z\right) + X^L\sin\left(r_z\right)\right)}{Z^L} \tag{A.29}$$

where for a single rotation r_z, the rotation matrix ${}^RR_L$ is represented as:

$${}^RR_L = \begin{bmatrix} \cos\left(r_z\right) & -\sin\left(r_z\right) & 0 \\ \sin\left(r_z\right) & \cos\left(r_z\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.30}$$

The vertical variation due to a small rotation δ applied to r_z is given by the following equation:

$$d_v = \left|v^R\left(\delta, X^L, Y^L, Z^L\right) - v^R\left(0, X^L, Y^L, Z^L\right)\right| = \left|\frac{f_y\left(Y^L - Y^L\cos\left(\delta\right) - X^L\sin\left(\delta\right)\right)}{Z^L}\right| \tag{A.31}$$

Rewriting the equation using the normalized coordinates x^L and y^L, as done before, we end-up with:

$$d_v = \left|f_y\left(y^L - y^L\cos\left(\delta\right) - x^L\sin\left(\delta\right)\right)\right| \tag{A.32}$$

This expression is of the form $|K|$ where one of the following conditions must be true for our case: i) $K > \mathcal{E}$ or ii) $K < -\mathcal{E}$ with $\mathcal{E}\in\mathbb{N}_{>0}$ representing the variation in pixel units. We will apply these two conditions to expression (A.32) to determine the optimal operation conditions.

For the case i), where $K > \mathcal{E}$ the boundary conditions expression is given by:

$$x^L < \frac{f_y y^L - f_y y^L\cos\left(\delta\right) - \mathcal{E}}{f_y\sin\left(\delta\right)} \tag{A.33}$$

This condition gives the first set of boundaries for x^L. Depending on the sign of δ, the condition creates a left or right boundary for x^L, that is slightly affected by y^L. Due to y^L effect, the boundaries are not completely vertical lines but slightly rotated.

If $\delta > 0$ the condition creates a left boundary with the best points being those closer to the bottom margin of the image, where y^L is maximum. If $\delta < 0$, the condition creates a right boundary again with the points closer to the bottom margin of the image generating a larger vertical displacement d_v.

For the case ii), where $K < -\mathcal{E}$ the boundary conditions expression is given by:

$$x^L > \frac{\mathcal{E} + f_y y^L - f_y y^L \cos(\delta)}{f_y \sin(\delta)} \tag{A.34}$$

This other condition gives another set of boundaries for x^L that again depend on the sign of δ. If $\delta > 0$ the condition creates a right boundary with the best points being those closer to the top margin of the image, where y^L is minimum. If $\delta < 0$ the condition creates a left boundary again with the points closer to the top margin of the image generating a larger vertical displacement d_v. From these two cases we conclude d_v is only greater than $\mathcal{E}$ if the chosen points are close to the left and right borders of the image.

Bibliography

[1] R. Beira, M. Lopes, M. Praca, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarn. "Design of the robot-cub (iCub) head". In: *IEEE International Conference on Robotics and Automation, ICRA 2006* (May 2006).

[2] A. Berthoz. In: *The sense of movement*. Harvard University Press, 2002.

[3] J.Y. Bouguet. *Camera Calibration Toolbox for Matlab*. 2008.

[4] G. Bradski. "OpenCV - Open Source Computer Vision". In: *Dr. Dobb's Journal of Software Tools* (2000).

[5] Kai Briechle and Uwe D. Hanebeck. "Template matching using fast normalized cross correlation". In: vol. 4387. 2001, pp. 95–102.

[6] Ruben Cantin-Martinez, Manuel Lopes, and Luis Montesano. "Body Schema Acquisition through Active Learning". In: *IEEE International Conference on Robotics and Automation*. Alaska, USA, 2010.

[7] John J. Craig. *Introduction to Robotics: Mechanics and Control*. 2nd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN: 0201095289.

[8] Edoardo Datteri, Giancarlo Teti, Cecilia Laschi, Guglielmo Tamburrini, Paolo Dario, and Eugenio Guglielmelli. "Expected perception: an anticipation-based perception-action scheme in robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1 (2003), pp. 934–939.

[9] Sean Ryan Fanello, Ugo Pattacini, Ilaria Gori, Vadim Tikhanoff, Marco Randazzo, Alessandro Roncone, Francesca Odone, and Giorgio Metta. "3D stereo estimation and fully automated learning of eye-hand coordination in humanoid robots". In: *14th IEEE-RAS International Conference on Humanoid Robots* (2014), pp. 1028–1035.

[10] K. A. Fleming, R. A. Peters II, and B. Bodenheimer. "Image Mapping and Visual Attention on a Sensory Ego-Sphere". In: *Proceddings IEEE/RSJ Conference on Intelligent Robotic Systems (IROS)*. 2006.

[11] Borko Furht and Oge Marques. *Handbook of Video Databases: Design and Applications*. 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2003. ISBN: 084937006X.

[12] Y. Furukawa and J. Ponce. "Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment". In: *International Journal of Computer Vision* 84.3 (2009), pp. 257–268.

[13] Andrea Fusiello, Emanuele Trucco, Alessandro Verri, and Ro Verri. *A Compact Algorithm for Rectification of Stereo Pairs*. 1999.

[14] Muscolo G., Recchiuto T., Hashimoto K., Dario P., and Takanishi A. "Towards an Improvement of the SABIAN Humanoid Robot: from Design to Optimization". In: *Journal of Mechanical Engineering and Automation*. 2012.

[15] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. "Online Continuous Stereo Extrinsic Parameter Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 17.2 (2012), pp. 234–238.

[16] Chris Harris and Mike Stephens. "A Combined Corner and Edge Detector". In: *The Plessey Company* (1988), pp. 147–152.

[17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[18] Richard I. Hartley. "In Defense of the Eight-Point Algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997).

[19] Lionel Heng, Bo Li, and Marc Pollefeys. "A Multiple-Camera System Calibration Toolbox Using A Feature Descriptor-Based Calibration Pattern". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (2013).

[20] Micha Hersch, Eric L. Sauser, and Aude Billard. "Online Learning of the Body Schema". In: *I. J. Humanoid Robotics* 5.2 (2008), pp. 161–181.

[21] Heiko Hirschmuller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.2 (2008), pp. 328–341.

[22] Matej Hoffmann, Hugo Marques, Alejandro Hernandez Arieta, Hidenobu Sumioka, Max Lungarella, and Rolf Pfeifer. "Body schema in robotics: a review". In: *IEEE Trans. Auton. Mental Develop.* 2.4 (2010), pp. 304–324.

[23] D. Hostetler and E. Larson. *Precalculus: A Concise Course*. Houghton Mifflin Co., 1994. ISBN: 0-618-62719-7.

[24] Moritz Knorr, Wolfgang Niehsen, and Christoph Stiller. "Online Extrinsic Multi-Camera Calibration Using Ground Plane Induced Homographies". In: *IEEE Intelligent Vehicles Symposium* (2013).

[25] Cecilia Laschi, Gioel Asuni, Eugenio Guglielmelli, Giancarlo Teti, Roland Johansson, Hitoshi Konosu, Zbigniew Wasik, Maria Chiara Carrozza, and Paolo Dario. "A bio-inspired predictive sensory-motor coordination scheme for robot reaching and preshaping". In: *Autonomous Robots* 25.1 (2007), pp. 85–101.

[26] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (2004), pp. 91–110. ISSN: 0920-5691.

[27] Giorgio Metta, Giulio Sandini, David Vernon, Lorenzo Natale, and Francesco Nori. "The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition". In: *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*. PerMIS '08. 2008, pp. 50–56.

[28] RC. Miall, DJ. Weir, Daniel M. Wolpert, and JF. Stein. "Is the cerebellum a smith predictor?" In: *Journal of Motor Behavior* 25 (1993), pp. 203–216.

[29] Nuno Moutinho, Martim Brandao, Ricardo Ferreira, Jose Antonio Gaspar, Alexandre Bernardino, Atsuo Takanishi, and Jose Santos-Victor. "Online calibration of a humanoid robot head from relative encoders, IMU readings and visual data". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (2012).

[30] Nuno Moutinho, Nino Cauli, Egidio Falotico, Ricardo Ferreira, José António Gaspar, Alexandre Bernardino, José Santos-Victor, Paolo Dario, and Cecilia Laschi. "An expected perception architecture using visual 3D reconstruction for a humanoid robot". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011), pp. 4826–4831.

[31] Nuno Moutinho, Ricardo Ferreira, José António Gaspar, Alexandre Bernardino, and Santos-Victor. "Markerless online stereo calibration for a humanoid robot". In: *4th International Conference on Development and Learning and on Epigenetic Robotics* (2014), pp. 454–460.

[32] Endo N., Momoki S., Zecca M., Saito M., Mizoguchi Y., ItohK., and A. Takanishi. "Development of whole-body emotion expression humanoid robot". In: *IEEE International Conference Robotics and Automation, ICRA*. 2008.

[33] Moreno P., Nunes R., Figueiredo R., Ferreira R., Bernardino A., Santos-Victor J., Beira R., Vargas L., Aragão D., and Aragão M. "Vizzy: A Humanoid on Wheels for Assistive Robotics". In: *ROBOT'2015 - Second Iberian Robotics Conference ROBOT2015*. Vol. 1. Nov. 2015, pp. 17–28.

[34] Pawel Pelczynski and Bartosz Ostrowski. "Automatic Calibration of Stereoscopic Cameras in an Electronic Travel Aid for the Blind". In: *Metrology and Measurement Systems* 20 (2013), pp. 229–238.

[35] Richard Alan Peters, Kimberly A. Hambuchen, and Robert E. Bodenheimer. "The sensory ego-sphere: a mediating interface between sensors and cognition". In: *Autonomous Robots* 26.1 (2008), pp. 1–19.

[36] M. Ribeiro. "Kalman and Extended Kalman Filters: Concept, Derivation and Properties". In: *Robotics WEBook, Instituto de Sistemas e Robotica, Instituto Superio Tecnico, Technical Notes* (2004).

[37] Rodrigues. "Des lois geometriques qui regissent les deplacements d'un système solide dans l'espace, et de la variation des coordonnees provenant de ces deplacements consideres independamment des causes qui peuvent les produire." In: *Journal de Mathematiques Pures et Appliquees* (1840), pp. 380–440.

[38] J. Santos, A. Bernardino, and J. Santos-Victor. "Sensor-Based Self-Calibration of the iCub's Head". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010* (2010).

[39] Jianbo Shi and Carlo Tomasi. "Good Features to Track". In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. 1994, pp. 593–600.

[40] S. Soatto. "A geometric framework for dynamic vision". PhD thesis. California Institute of Technology, 1996.

[41] S. Soatto, R. Frezza, and P. Perona. "Motion estimation via dynamic vision". In: *In IEEE Transactions on Automatic Control* (1996).

[42] B. Tworek, A. Bernardino, and J. Santos-Victor. "Visual self-calibration of pan-tilt kinematics structures". In: *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, ROBOTICA 2008* (April 2008).

[43] Pedro Vicente, Lorenzo Jamone, and Alexandre Bernardino. "Online body schema adaptation based on internal mental simulation and multisensory feedback". In: *Frontiers in Robotics and AI* 3.7 (2016). ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00007.

[44] Michael Warren, David McKinnon, and Ben Upcroft. "Online Calibration of Stereo Rigs for Long-Term Autonomy". In: *IEEE International Conference on Robotics and Automation (ICRA)* (2013).

[45] Thomas P. Webb, Richard J. Prazenica, Andrew J. Kurdila, and Rick Lind. "Vision-Based State Estimation for Autonomous Micro Air Vehicles". In: *Journal of Guidance, Control, and Dynamics* 30 (2007), pp. 816–826.

[46] Kai Welke, Markus Przybylski, Tamim Asfour, and Rudiger Dillmann. "Kinematic Calibration for Saccadic Eye Movements". In: *Technical Report, U. Karlsruhe* (2008).

[47] D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan. "Perspectives and problems in motor learning". In: *Trends in Cognitive Sciences* 5 (2001), pp. 487–494.

[48] D.M. Wolpert, D.J. Miall, and M. Kawato. "Internal Models in the celebellum". In: *Trends in Cognitive Sciences*. Vol. 2(9). 1998, pp. 338–347.