RoboCup SS17

submitted
PRACTICAL COURSE

B.Sc. Tianming Qiu

born 30.03.1993
resident in:
Felsennelkenanger 15
80937, Munich
Tel.: +49 17643387325

Institute for
CONGNITIVE SYSTEMS
Technical University of Munich

Prof. Dr. Gordon Cheng

Supervisor:	Mohsen Kaboli
Start:	24.04.2017
Delivery:	24.07.2017

Abstract

RoboCup is a competition for soccer robot held annually and we work on the RoboCup Standard Platform League (SPL) by using humanoid robot NAO, based on the code framework from University of Bremen's team B-Human.

Self-localization is a very important task for NAOs during the match. The action decisions, ball tracking and different team cooperation strategies are dependent on the precise location on the standard soccer field. Self-localization is also related to computer vision and statistical signal processing. These topics are the foundations of autonomous robot. So this semester I focus on figuring it out, how does NAO localize itself on the soccer field by cameras and how could it be improved.

The current method in B-Human framework has been continuously developed, verified and implemented for several years. It has been considered as state-of-the-art method which is already very robust and compact. The main idea of self-localization in B-Human framework is to apply particle filter with unscented Kalman filter for each particle to estimate robot pose. But there are still some issues that when robot is walking, some shaking noises from camera will cause deviations in visual odometry. PnP pose estimate would be adopted instead of relative distance calculation, which would take all the transforms generally into consideration and eliminate the error.

Contents

1 Introduction 5
1.1 Motivation 5
1.2 My Work 5

2 Self-Localization 7
2.1 General Framework: Particle Filter 7
2.2 Combination with Unscented Kalman Filter 9

3 Improvement of Visual Odometry 11
3.1 The current method 11
3.1.1 Visual odometry procedure 11
3.1.2 Current problem 12
3.2 PnP Pose estimation 14

4 Conclusion 15
4.1 Result and Comparison 15
4.2 Future Work 15
4.2.1 Come up with a more adequate noise model 15
4.2.2 Search for ball 15
4.2.3 Data association error 16
4.2.4 Communication with another robots 16

List of Figures 19

Bibliography 21

Chapter 1

Introduction

RoboCup is a competition for the soccer robot held annually, which aims at promoting researches on robots and artificial intelligence. We work on the RoboCup Standard Platform League (SPL) by using humanoid robot NAO. There are 5 players for each side: goalkeeper, defender, supporter and striker. Different roles share some same functions such as self-localization, tracking the ball, kicking and so on. They also have different strategies according to their distributed job on the field just like the real soccer match. Of course the necessary communication and cooperation are required to achieve better performance in the match. One of the most successful team is B-Human from University of Bremen, who have won the championship five times. Last year TU Munich also set up our own RoboCup team called “TUM Lion”.

1.1 Motivation

The course “Introduction Lab Humanoid RoboCup” in summer semester 2017 is given for the students are interested in RoboCup and gives them a brief idea about RoboCup as well as the general knowledge of practice on autonomous robot localization, computer vision, motion control and multi-robot cooperation tasks. This semester’s project is based on the code framework from University of Bremen’s team B-Human. With the help of the open source framework could let us start the project more quickly and do some specific research on the topic what we are interested in or more familiar with. At the same time, some issues will be found and could be improved or considered as another new strategy.

1.2 My Work

Since we have already got the access to the state-of-the-art open source framework, so the first stage is to get familiar with the procedure of NAO operating and the codes. It includes:

- Basic operations on NAO and remote control through SimRobot
- Calibration(including camera parameters, color and initial motions)
- Understand how the current framework realize the self-localization

The second stage is focusing on self-localization and finding some interesting part to improve. Firstly I learned the Markov localization theory of mobile robot such as particle filter and unscented Kalman filter, which is used in the B-Human framework. Then the PnP pose estimation method is adopted to eliminate the current visual odometry error. In Chapter 2 I will introduce how does the current method figure out the localization of robot and the procedure of particle filer and unscented Kalman filter. In Chapter 3 I will show the visual odometry procedure of B-Human and explain what kind of noise it could be as well as the method that I applied.

Chapter 2

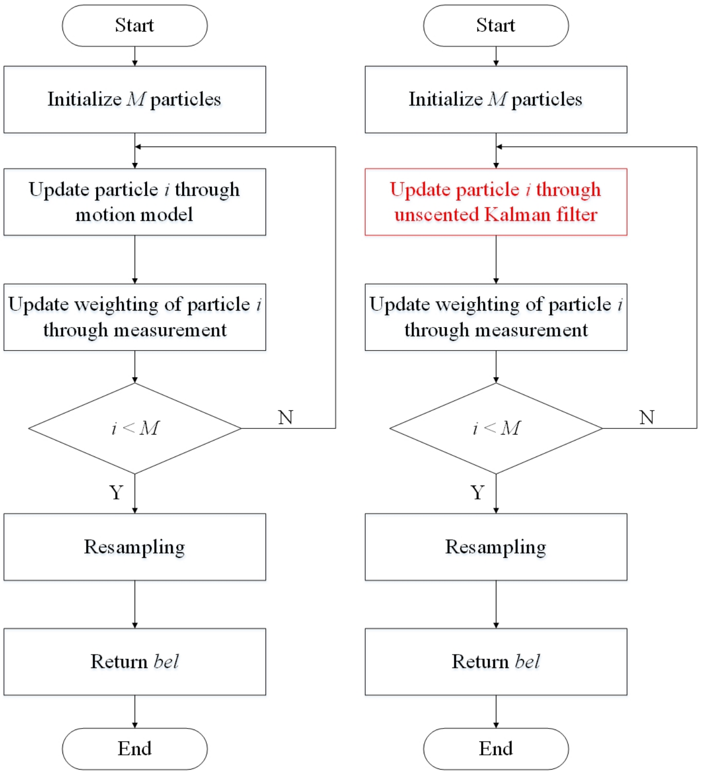
Self-Localization

This chapter introduces the concrete method used in the current framework for NAO, the mobile robot localization. RoboCup self-localization is a problem that NAO robot could determine its pose, which includes both position and angular, relative to the given map of the standard soccer field. Almost all the tasks for each robot player require knowledge of its pose on the field. We call this problem as rather a self-localization than a SLAM(Simultaneous localization and mapping) problem, one important reason is the map of the environment has been provided as prior knowledges. Self-localization could be treated as a problem of coordinate transformation. Map of the field or environment is described as the global coordinate system, which does not rely on robot's pose. The robot pose will be finally transformed into the global frame and represented as a global coordinate[1]. In our project the NAO pose is only changes in a plane, the z element is always remains as zero since the robot always stands firmly on the ground. So the pose could be written as $(x, y, \theta)^T$.

However, the pose of robot could not be acquired directly, instead it could be inferred from the sensor data, which here we read it from cameras. The sensor could not be hundred percentage accurate, it always contains noise, especially image processing by using camera or illumination changes or other possible visual noises. So we have to find a proper method to calculate the pose from imperfect sensor data, which will be introduced in this chapter.

2.1 General Framework: Particle Filter

For self-localization, B-Human uses a particle filter based on the Monte Carlo method [2] as it is a proven approach to provide more accurate results in such an environment [3]. The reason for selecting particle filter in this self-localization problem is that in the RoboCup soccer match the robot will from time to time fall down, and then it will not get the correct knowledge of its position. The issue is that the robot might believe that it knows where it is while it does not. This is called "kidnapped robot problem" [1]. So the particle filter will be adopted to solve


Start
Initialize M particles
Update particle i through motion model
Update weighting of particle i through measurement
$i < M$
N
Y
Resampling
Return bel
End
Start
Initialize M particles
Update particle i through unscented Kalman filter
Update weighting of particle i through measurement
$i < M$
N
Y
Resampling
Return bel
End

(a) Particle filter (b) combine PF and UKF

Figure 2.1: Self-localization based on particle filter flowchart

this kidnapped robot problem and it does not demand initial position of robot. The particle filter is a kind of Bayes filter, based on Monte Carlo method. It uses several samples as particles to describe the posterior. In particle filter, the samples of a posterior distribution are called particles are represented as [1]:

$$\mathcal{X}_t := x_t^{[1]}, x_t^{[2]} \dots x_t^{[M]}$$

The posterior of each particle $x_t^m (with 1 \leq m \leq M)$ is the probability of the state at specific time t. Usually, M, the number of the particles is very large to get a as perfect as possible approximation, since particle filter is a non-parametric method to estimate the state updating. However in the RoboCup situation, only 12 to 16 particles are adopted. Because the memory space and computation speed are both limited on the robot NAO. During the match, it has a high demanding on real time performance. Here I have not tested or verified the effectiveness of the so small number of samples. But in recent years B-Human team even drop the particles from 16 to 12 [4] and it is proven in practice which shows a great robustness and outstanding performance. Then how does robot NAO apply particle filter to localize itself is shown in the flowchart Figure 2.1(a).

2.2 Combination with Unscented Kalman Filter

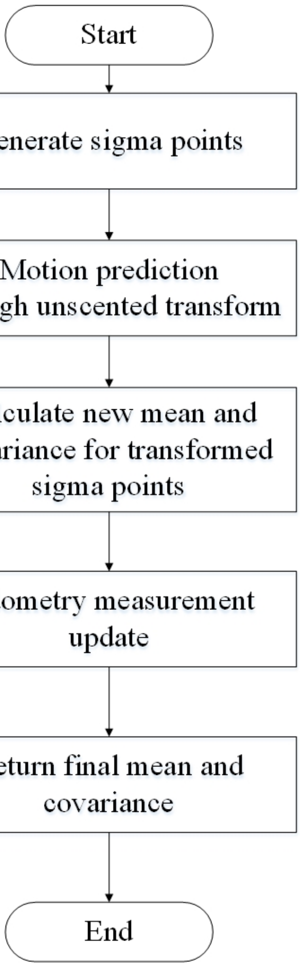

Start
Generate sigma points
Motion prediction through unscented transform
Calculate new mean and covariance for transformed sigma points
Odometry measurement update
Return final mean and covariance
End

Figure 2.2: Unscented Kalman filter

Since 2012, B-Human framework made a big change on self-localization: a combination of a particle filter and an Unscented Kalman filter is used. The former computes a global position estimate; the latter performs local tracking for refining the global estimate [5]. This implementation will improve the local accuracy of particle filter and further results in the improvement of localization precision. The change has shown in the Figure 2.1(b) .

The idea behind the Unscented Kalman filter is similar to the original Kalman filter. However it is applied for the non-linear motion model and provides preciser result than Extented Kalman filter which only uses first order approximation of Talor series. The idea of Unscented Kalman filter is that it generates sigma points and uses the unscented transform to describe the non-linearity. The mean value and covariance calculation is based on these sigma points. Since the state of pose could be represented as $(x, y, \theta)^T$, it contains three dimensions, then the number of sigma points for each state is $2n + 1 = 7$, here n is the dimension number of the state variable. Each sigma points will be predicted by motion model. And due to the non-linearity of motion model, unscented transform will be used. As shown in Figure 2.2, the mean as well as covariance of states will also be modified with measurement and finally be returned.

Generally, the localization process will use particle filter to estimate the robot pose and each particle will use unscented Kalman filter to update the pose of the sample.

Chapter 3

Improvement of Visual Odometry

Above the procedure of self-localization was described in general. Particle filter require sensor measurement to calculate the weighting for each particle and Unscented Kalman filter demands the sensor measurements to update the motion predictions. Visual odometry is a widely used with strong functions sensor measurement method, which is applied in this situation and provides the measurement for each kind of filter. So the accuracy of pose estimated by visual part will definitely have an obvious influence on the final result. This chapter will introduce what knowledge of the sensor measurement can provide and how the sensor measurement take place.

3.1 The current method

3.1.1 Visual odometry procedure

There are several sensors to acquire information from the environment on NAO: camera, microphone and sonar. From them only the camera is adopted, since the environment of the soccer match will be quite complicated, for example:

- The white border lines and green field are on the same plane.
- The goal posts are perhaps too far away from the robots sometimes.
- There are 6 moving robots on the field, which the opponents are distinguished from our teammates only by colors.
- The ball is small and the detection of a ball requires high precision.

So based on above reasons, the another sensors are not qualified for the requirements. However, the camera is suitable for each specific task and the computer vision skill as well as visual odometry are being perfect so far.
There two cameras on the NAO's head which can provide large FOV. Several tasks such as image preprocessing, feature detection(including: line perception, penalty

mark perception, ball perception) and data association will be performed on the image gathered by each camera. B-Human's current vision system provides a variety of perceptions that have all been integrated into the sensor model: goal posts (ambiguous as well as unambiguous ones), line segments (of which only the endpoints are matched to the field model), line crossings (of three different types: L, T, and X), and the center circle [5]. These features can be associated to the feature detected in pixel images. The detailed of these tasks will not be mentioned in my part of report, which are introduced by my teammates. My work is that suppose the the feature detection and data association have been finished.
Based on feature detection and data association, the corresponding point pairs have been already found. For example the goal post in figure i. whose coordinate in the homogeneous global 3D world could be represented as $(X, Y, 0, 1)^T$. And the corresponding homogeneous coordinate in pixel coordinate can be written as $(x, y, 1)^T$. Since the camera calibration matrix, including both intrinsic and extrinsic calibration parameters, has been acquired through camera calibration before each match. Assuming the camera model as pinhole camera model shown in Figure 3.1 [6]:

$$\mathbf{x} = \mathbf{K} \cdot \mathbf{X}_{cam}$$

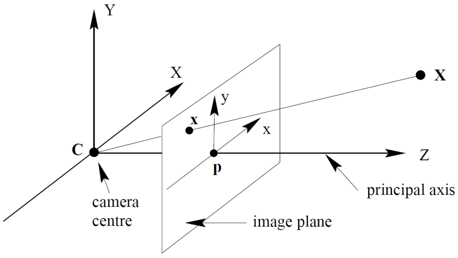
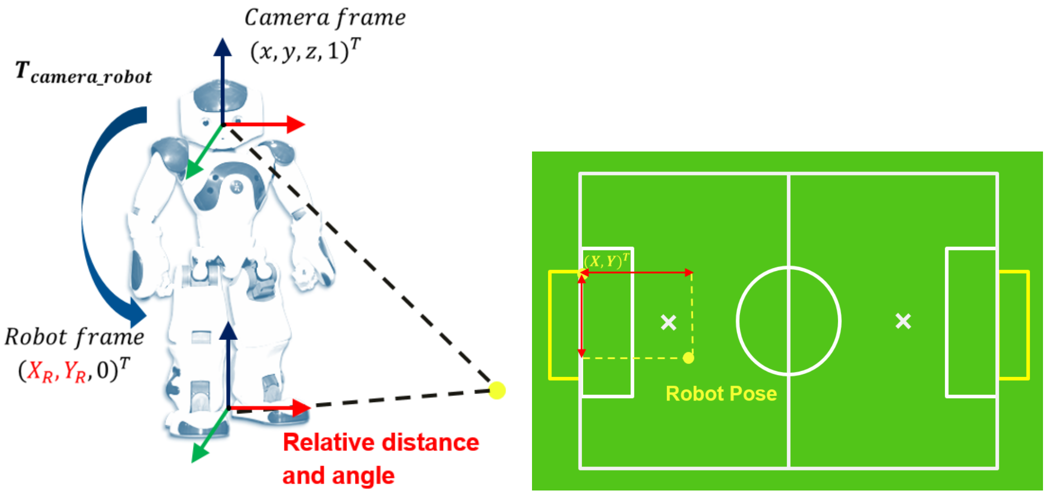


Figure 3.1: Pinhole camera model

So we could get the corresponding 3D goal post coordinate in the ideal camera frame. Since we know the geometrical value of NAO's body construction, we can transform the coordinates into the robot frame, which is shown in the Figure **??** Based on this, we could get the related distance between robot and the goal post in robot frame. At the same time, the precise location of goal post in the global field is also known. So we could calculate the robot pose from this above information.

3.1.2 Current problem

In the process of transforming the coordinate from camera frame into robot frame, the current method only use a constant transform. However when robot is walking, it cannot always be extremely vertical to the ground. The central line of its body and camera on the head will have an arbitrary angle of tilt. Also due to detecting of


Camera frame
$(x, y, z, 1)^T$
T_{camera_robot}
Robot frame
$(X_R, Y_R, 0)^T$
Relative distance and angle
$(X, Y)^T$
Robot Pose

(a) Coordinate transform (b) Relative pose to landmark

Figure 3.2: Pose estimation

the environment, robot always turn around its head or even look up and down, but this information will also create bias by coordinate transforming. Then it will cause a bias on the camera frame transform, which is shown in Figure 3.3. This error between the real robot base coordinate and the result calculated by the constant transform will further affect the pose estimation.

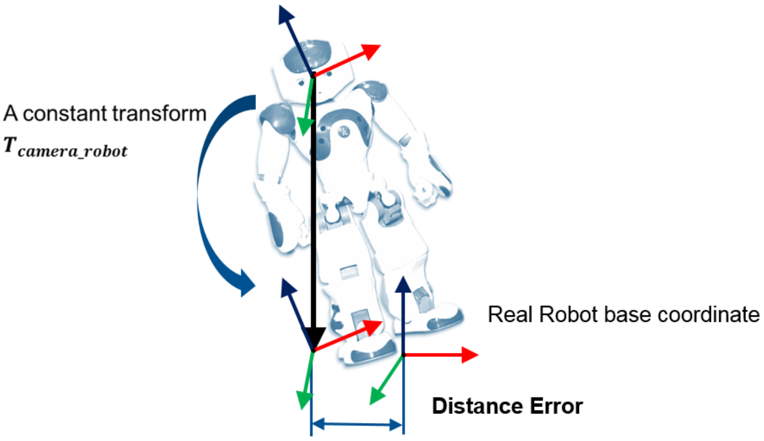

A constant transform
T_{camera_robot}
Real Robot base coordinate
Distance Error

Figure 3.3: Position error by a constant transform

3.2 PnP Pose estimation

Since the current method contains the tilting and shaking error I apply the PnP method [7] which could eliminate the error by considering the transform in a more direct way. Reconsidering the corresponding point pair $\mathbf{x} := (x, y, 1)^T$ in pixel coordinate and $\mathbf{X} := (X, Y, 0, 1)^T$ in 3D global space. They should have this following relationship (3.1):

$$\mathbf{x} = \mathbf{T}_{image2field} \cdot \mathbf{X}_{field} \tag{3.1}$$

And the transform matrix $\mathbf{T}_{image2field} \in \Re^{3\times 4}$ could be represented as three parts:

$$\mathbf{T}_{image2field} = \mathbf{K} \cdot \mathbf{T}_{camera2robot} \cdot \mathbf{T}_{robot2field} \tag{3.2}$$

In (3.2) what we can figure out from the corresponding point pairs is $\mathbf{T}_{image2field}$, and what we want to know is $\mathbf{T}_{robot2field}$. If we know the $\mathbf{T}_{robot2field}$ then we could directly read robot pose from this matrix. Also, the camera calibration matrix $\mathbf{K}$ is also already known because the calibration has been done before each match. $\mathbf{T}_{robot2field}$ can also be represented as (3.3):

$$\mathbf{T}_{robot2field} = \begin{bmatrix}\mathbf{R} & \mathbf{T}\end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha & 0 & T_x \\ -sin\alpha & cos\alpha & 0 & T_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.3}$$

There are 4 unknown parameters in (3.3), so at least 2 pairs of corresponding points are required to solve the equation. The 2 pairs of corresponding points can be selected by two goal posts or the start and end point of a specific line. Suppose $\mathbf{M} := \mathbf{K} \cdot \mathbf{T}_{camera2robot}$, elements in $\mathbf{M}$ are all known values, so the equation can be written as (3.4):

$$\mathbf{x} = \mathbf{M} \cdot \begin{bmatrix}\mathbf{R} & \mathbf{T}\end{bmatrix} \cdot \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \tag{3.4}$$

$$[\mathbf{x_1} \quad \mathbf{x_2}] = \mathbf{A} \cdot [\mathbf{X_2} \quad \mathbf{X_2}] \tag{3.5}$$

So, in (3.5) it shows the how does two pairs of corresponding points calculate matrix $\mathbf{A}$ which contains four unknown parameters. If we have more than 4 pairs of corresponding points, SVD method for minimum solution finding could be applied.

Chapter 4

Conclusion

4.1 Result and Comparison

Before the robot always gets lost and cannot go to the assigned area, such as defending area for defender even at the beginning of match. After using PnP pose estimation method, the accuracy of pose calculation has been improved. The self-localization process is much more quick and robust. Averagely, the robot NAO could go to the specific position from initial position out of the border in 30 seconds. The current method which based on only one pair of corresponding points will be less computation cost but it does not consider the tilting pose of the robot body and rotating of the robot head as well as its camera. However the precise localization of robot is more important and the result will be fed to both unscented Kalman filters and particle filter, which will further affect the self-localization.

4.2 Future Work

4.2.1 Come up with a more adequate noise model

This semester we focused on the visual odometry part, just the sensor measurement of the both two filter method. The motion model also plays an important role on the localization and is restricted with noise comes from actuators of each joint as well as the ground friction at the same time. So in the future, this part of work should be done and set up a more proper and scientific noise model to describe the uncertainty of motion model.

4.2.2 Search for ball

There is still some problem of searching for the ball, this part belongs not to the localization problem, but a tracking problem.

4.2.3 Data association error

According to the both current method and PnP pose estimation method, the correctness of data association determines the final quality of self-localization directly. Whether the current data association robust enough and which noise comes from, these issues should be more discussed.

4.2.4 Communication with another robots

This semester we only operate on each single robot without communication with another team players. The communication is very useful and it can gather more information and more fast. For example, if one of the robot has seen the ball, it could ‘tell’ other robot where the ball is. Then the other robot will not implement the ball search task any more and do another more efficient job. Furthermore, if two robot detect the ball at the same time, they could make a data fusion through communication to improve the accuracy.

Acknowledgments

First of all, I would like to express my thanks to Prof. Dr. Gordon Cheng and Chair of Cognitive Systems for offering this course and all the hardware equipments.
I would further like to thank our supervisor Mohsen Kaboli for your patient help and many constructive suggestions. You are always willing to share your experiences and give us significant guides. Your office is always open to us. You are strict to us but I really learn a lot from that. Not only the experiment itself also the way how to deliver a presentation.
Also, I would appreciate the help offered by our teaching assistant Zhiliang Wu, who prepared all the lab stuff for us and wrote detailed tutorials that let me know how to operate NAO step by step. Your careful work makes me to get start quickly as soon as possible and gives me a lot confidence to finish the task.
I am particularly thankful to my team mate: Yao, Fabian, Zhiyi, Jingjie, and Minkai. We discussed a lot and you always give me supports.

List of Figures

2.1 Self-localization based on particle filter flowchart 8
2.2 Unscented Kalman filter 9

3.1 Pinhole camera model 12
3.2 Pose estimation 13
3.3 Position error by a constant transform 13

Bibliography

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* 2005.

[2] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[3] Thomas Röfer, Tim Laue, and Dirk Thomas. Particle-filter-based self-localization using landmarks and directed lines. pages 608–615. Springer, 2005.

[4] Thomas Röfer, Tim Laue, Judith Müller, Armin Burchardt, Erik Damrose, Alexander Fabisch, Fynn Feldpausch, Katharina Gillmann, Colin Graf, Thijs Jeffry de Haas, Alexander Härtl, Daniel Honsel, Philipp Kastner, Tobias Kastner, Benjamin Markowsky, Michael Mester, Jonas Peter, Ole Jan Lars Riemann, Martin Ring, Wiebke Sauerland, André Schreck, Ingo Sieverdingbeck, Felix Wenk, and Jan-Hendrik Worch. B-human team report and code release 2010, 2010. Only available online: `http://www.b-human.de/downloads/bhuman10_coderelease.pdf`.

[5] Thomas Röfer, Tim Laue, Judith Müller, Michel Bartsch, Malte Jonas Batram, Arne Böckmann, Nico Lehmann, Florian Maaß, Thomas Münder, Marcel Steinbeck, Andreas Stolpmann, Simon Taddiken, Robin Wieschendorf, and Danny Zitzmann. B-human team report and code release 2012, 2012. Only available online: `http://www.b-human.de/wp-content/uploads/2012/11/CodeRelease2012.pdf`.

[6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[7] Simon Flückiger. Self-localization on nao robots based on external features: Pnp pose estimation, 2016. Only available online: `https://drive.google.com/file/d/0B8OEfzrd1JHkbkNqMkJYT0tZbUk/view`.