

Automatic Robot Calibration for the NAO

Tobias Kastner¹ (✉), Thomas Röfer², and Tim Laue¹

¹ Fachbereich 3 – Mathematik und Informatik, Universität Bremen,
Postfach 330 440, 28334 Bremen, Germany
{[dyeah](mailto:dyeah@informatik.uni-bremen.de),[tlaue](mailto:tlaue@informatik.uni-bremen.de)}@informatik.uni-bremen.de

² Deutsches Forschungszentrum Für Künstliche Intelligenz, Cyber-Physical Systems,
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
thomas.roefer@dfki.de

Abstract. In this paper, we present an automatic approach for the kinematic calibration of the humanoid robot NAO. The kinematic calibration has a deep impact on the performance of a robot playing soccer, which is walking and kicking, and therefore it is a crucial step prior to a match. So far, the existing calibration methods are time-consuming and error-prone, since they rely on the assistance of humans. The automatic calibration procedure instead consists of a self-acting measurement phase, in which two checkerboards, that are attached to the robot's feet, are visually observed by a camera under several different kinematic configurations, and a final optimization phase, in which the calibration is formulated as a non-linear least squares problem, that is finally solved utilizing the *Levenberg-Marquardt* algorithm.

1 Introduction

Calibration is the process of determining the relevant parameters of a robotic system by comparing the prediction of the system's mathematical model with the measurement of a known feature, which is considered as the *standard*. If the difference between the prediction and the measurement exceeds a certain tolerance, it is inevitable to compensate for this mismatch to allow the robot to operate as desired. The automatic calibration method presented in this paper is customized for the humanoid robot NAO [4] from *Aldebaran Robotics*. The NAO has 21 degrees of freedom and is equipped with two cameras. The images are taken at a frequency of 30 Hz by each camera while other sensor data (such as joint angle measurements) are updated at 100 Hz. The operating system is an embedded Linux that is powered by the *Intel-Atom* processor at 1.6 GHz. Since 2008, the NAO is the official platform of the RoboCup Standard Platform League (SPL). To play soccer, a NAO has to be able to walk fast and to score goals. The overall performance of these two essential tasks strongly depends on prior precise calibration, which, so far, is a manual procedure that requires the assistance of human users. It is time-consuming and error-prone, since the errors in the kinematics are estimated by eye and hand. These errors arise from the imprecise assembly of the robot itself and damage and wearout of the motors

during its operation. Consequently, it is often necessary to recalibrate a NAO after a match. The goal of the automatic calibration that is presented in this paper is to reduce the workload of human users, while providing comparable appropriate calibration parameters.

Markowsky [8] developed a method to calibrate the NAO’s leg kinematics by using a *hill climbing* algorithm and a cost function that computes the distribution of the current and weight loads of the two legs. The assumption was that the NAO is calibrated if the loads of each leg are similar. Due to poor sensor readings, this method turned out to be inapplicable. This paper is rather inspired by the works of Birbach *et al.* [2], Pradeep *et al.* [11], and Nickels [10]. These methods are based on the *hand-eye calibration*, which is explained by Strobl *et al.* [13]. The goal of these calibration procedures is to find the poses of the cameras in the robot’s head as well as the joint offsets that affect the proper positioning of the robot’s end effectors. The robot’s end effectors are driven to pre-recorded positions and markers attached to these end effectors are visually detected by the robot’s cameras. Each position of the marker’s features in pixel coordinates, as well as the respective joint angle measurements of the kinematics are gathered and finally used to minimize the deviation between the actual visual measurements and the predicted ones based on the mathematical model of the robot, i. e. forward kinematics and extrinsic and intrinsic camera parameters.

In this paper, the NAO is calibrated by moving two checkerboards, which are attached to the robot’s feet, in front of the lower camera of the NAO’s head. Similar to the related work presented, the image position of each vertex (point of contact of two isochromatic tiles) on the checkerboard is measured together with the current joint angles. To formulate the calibration as a problem of non-linear least squares, the forward kinematics of the NAO are computed with homogeneous coordinate transformation and the projection function of the cameras is described as a pinhole camera model. Using an a-priori intrinsic camera calibration and the *Levenberg-Marquardt* algorithm, the 21 parameters involved (12 joint offsets, 3 camera rotation errors and 6 correction parameters for imprecisely mounted checkerboards) are estimated by minimizing the sum of squared residuals of each measured vertex in pixels and the projection of the corresponding vertex from robot-relative coordinates to 2D image points, utilizing the joint angles measured.

The remainder of this paper is structured as follows: in the next section, the properties of the checkerboard and foot-assembly as well as the formulation of the calibration as a problem of non-linear least squares are discussed. Section 3 presents the results achieved, followed by Sect. 4, which concludes the paper and gives an outlook on possible future work and improvements to the calibration procedure presented.

2 Automatic Robot Calibration

First of all, the properties of the checkerboard sandals are discussed, followed by some implementation details and the definition of the calibration as a non-linear least squares problem.

2.1 Checkerboard Sandals

The checkerboard pattern with 35 tiles (7 columns and 5 rows) is printed on a customized aluminum composite panel (see Fig. 1b) with a square size of 2.5 cm. Each board is mounted with four pins that exactly fit into the recesses located at the force-sensing resistors (FSR) on the bottom of the NAO's feet (soles) and a further fixture with a hook and loop fastener tape. Since the positions of the FSRs (see Fig. 1a) are known relative to the projection of the ankle joint on the sole, the positions of all 24 checkerboard vertices can be computed easily, considering that the center of the checkerboard is situated 17 cm away from the joint projection mentioned. These positions are used to predict the image coordinates of each vertex using the mathematical model of the NAO that still has to be defined.

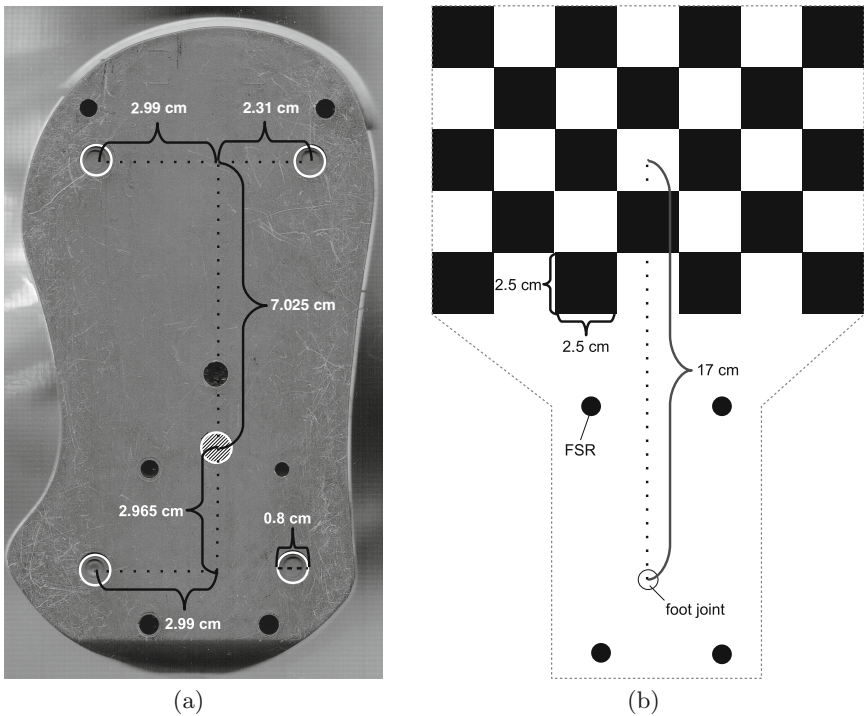


Fig. 1. The positions of the FSRs on the sole (a). The checkerboard sandal with some dimension information (b).

2.2 Implementation

The calibration is implemented using the *C++* framework [12] for modular development of robot control programs, published by the SPL team *B-Human*. In the context of this paper, a motion control engine, a checkerboard recognizer, an

optimization algorithm, and a calibration control module were developed. Any more in-depth information on those software components can be found in the corresponding thesis [5].

As proposed by Wiest [15] and Elatta *et al.* [3], the calibration is subdivided into four phases *modeling*, *measurement*, *identification*, and *compensation*.

Modeling. The conversion from world to image coordinates is modeled with the pinhole-camera model P :

$$P(x, y, z) = O_C - (y, z)^T \frac{f_l}{x}. \quad (1)$$

The optical center O_C and the focal length f_l were determined in a prior intrinsic camera calibration, as proposed by Zhang [16].

The camera-relative world coordinates x , y , and z are computed using a measured set of joint angles q , the forward kinematics $T_{LS}^O(q)$, the transformation from the ankle joint to the left sole, and the positions of the checkerboard vertices computed relative to the feet $p_{LS\alpha}$. The functions Rot_x , Rot_y , and Rot_z are used to rotate a pose (position and rotation) around the respective axis, named in the suffix. Likewise, the functions $Trans_x$, $Trans_y$, and $Trans_z$ translate a pose along the respective axis. In Fig. 2, the kinematic tree of the NAO is shown.

Each joint angle q_i is composed from the actual measured angle θ_i and the affecting offset α_i :

$$q_i = \theta_i + \alpha_i, \quad i \in \left\{ LHipYawPitch, LHipRoll, LHipPitch, \right. \\ \left. LKneePitch, LAnklePitch, LAnkleRoll, \right. \\ \left. RHipYawPitch, RHipRoll, RHipPitch, \right. \\ \left. RKneePitch, RAnklePitch, RAnkleRoll \right\}. \quad (2)$$

Each element i stands for a servo motor that is able to measure its current angle with a precision of 0.1° (the same precision applies to commanded angles). The position of each motor is visualized in Fig. 2.

The robot-relative pose of the lower camera is computed with T_C^O (homogeneous transformation from the robot origin to the camera):

$$T_C^O(q, \alpha) = T_C^O(q) \cdot Rot_y(\alpha_{CamPitch}) \cdot Rot_x(\alpha_{CamRoll}) \cdot Rot_z(\alpha_{CamYaw}). \quad (3)$$

It is assumed that the actual camera pose $T_C^O(q)$ is affected by three rotational offsets $\alpha_{CamPitch}$, $\alpha_{CamRoll}$, and α_{CamYaw} .

With the information on the checkerboard given in Sect. 2.1, the coordinates of all vertices p_{LS} , relative to the soles, can be computed.

$$p_{LS\alpha} = Rot_z(\alpha_{LRotZ}) \cdot Trans_x(\alpha_{LTransX}) \cdot Trans_y(\alpha_{LTransY}) \cdot p_{LS}. \quad (4)$$

Since the pins on the boards are glued, it is possible that the assembly on the soles is imprecise. In addition to the joint and camera correction offsets, three

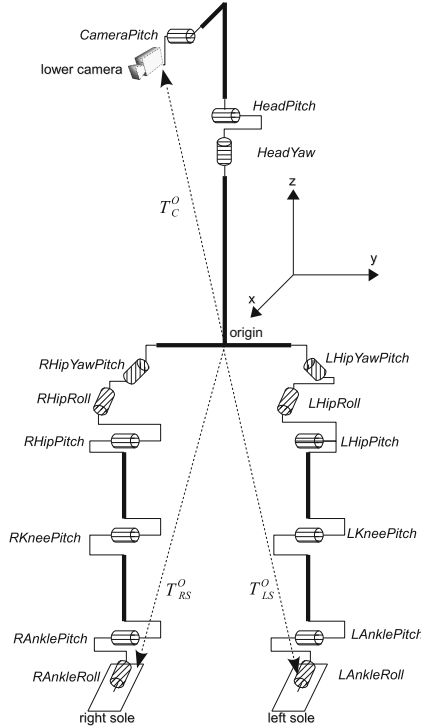


Fig. 2. The kinematic tree of the NAO. The origin is located between the two hip motors. The *CameraPitch* is a fixed assembly angle, taken from the official NAO documentation. Any other motor angles are modifiable.

further parameters for each board are considered. It is assumed that the boards have a rotational offset α_{LRotZ} around the z -axis and a translational error in x and y direction ($\alpha_{LTransX}$ and $\alpha_{LTransY}$).

The final model is defined with $proj(p_{LS}, \theta, \alpha)$:

$$proj(p_{LS}, \theta, \alpha) = P(T_C^O(q, \alpha)^{-1} \cdot T_{LS}^O(q) \cdot p_{LS\alpha}). \quad (5)$$

A given sole-relative vertex coordinate is transformed into origin coordinates, which gets further converted into camera-relative coordinates and is finally projected onto the NAO's camera plane, resulting in a prediction, where a vertex should be located in the image, considering the current joint data and the calibration parameters.

The definitions given above are for the left leg kinematics. The missing projection function for the right leg is computed analogously, using the right leg's joint data and the vertex positions from the right checkerboard. Also, only the leg's kinematics are adjusted, since, so far, there are no actions that require precisely calibrated arms.

Table 1. Listing of all identified parameters. A type of Δ° indicates a rotational offset and a Δmm a translational offset.

Parameter	Type	Parameter	Type	Parameter	Type
α_{CamYaw}	Δ°	$\alpha_{CamPitch}$	Δ°	$\alpha_{CamRoll}$	Δ°
$\alpha_{LHipYawPitch}$	Δ°	$\alpha_{LHipRoll}$	Δ°	$\alpha_{LHipPitch}$	Δ°
$\alpha_{LKneePitch}$	Δ°	$\alpha_{LAnklePitch}$	Δ°	$\alpha_{LAnkleRoll}$	Δ°
$\alpha_{RHipYawPitch}$	Δ°	$\alpha_{RHipRoll}$	Δ°	$\alpha_{RHipPitch}$	Δ°
$\alpha_{RKneePitch}$	Δ°	$\alpha_{RAnklePitch}$	Δ°	$\alpha_{RAnkleRoll}$	Δ°
α_{LRotZ}	Δ°	$\alpha_{LTransX}$	Δmm	$\alpha_{LTransY}$	Δmm
α_{RRotZ}	Δ°	$\alpha_{RTransX}$	Δmm	$\alpha_{RTransY}$	Δmm

Altogether, the vertex-projection model is affected by 21 parameters (see Table 1), that need to be adjusted. The crucial parameters are the three rotational camera corrections and the twelve leg joint offsets. Note that this set of parameters is the same that is used in the manual calibration procedures.

Measurement. The target measurements are the 2D image coordinates of the checkerboard vertices. The localization of these vertices is accomplished with a combination of the vertex detection algorithm of Bennet *et al.* [1], the chessboard extraction method of Wang *et al.* [14], and the sub-pixel refinement implemented by Birbach *et al.* [2]. These methods operate on the grey-scale images of the lower camera with VGA resolution.

To assure that the calibration can be performed secure and unsupervised, the NAO is lying on its back, while driving to 24 different leg stances (for both legs). For each configuration, the checkerboard is observed incorporating three different head postures. The configurations are chosen heuristically, considering that the checkerboard is fully visible, while having different rotations and distances to the camera. Note that the checkerboard vertices are only detected after the robot reached its desired stance, since there is a lag between the arrival of a new camera image and the sensor readings. The complete self-acting measurement phase is visualized in Fig. 3.

For each vertex observed, a new measurement m is collected and added to a set M :

$$M = \{m_0, m_1, \dots, m_n\}. \quad (6)$$

A measurement is a triple, consisting of the image position p_{img} , the corresponding sole-relative 3D coordinate of the vertex p_{cb} , and the measured joint angles θ :

$$m = (p_{img}, p_{cb}, \theta). \quad (7)$$

Identification. After the completion of the measurement phase, the optimization procedure is initialized. First of all, the set of residuals R is built, calculating

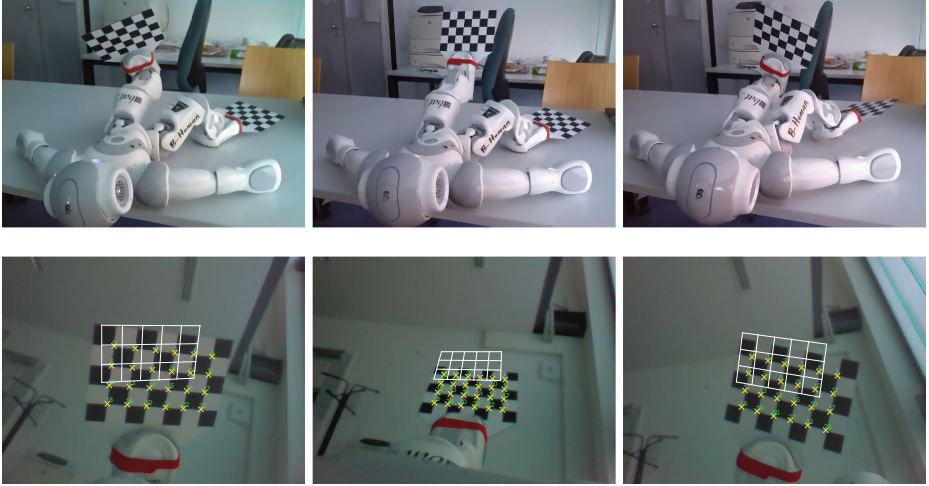


Fig. 3. Some snapshots taken during a calibration experiment. The upper images show different joint angle configurations, whereas the lower figures depict the corresponding points of view of the lower camera.

the deviation of each measurement's image coordinate $m[p_{img}]$ and the corresponding prediction, using the projection model (see Eq. 5) and the measurement's joint data $m[\theta]$, the sole-relative vertex position $m[p_{cb}]$, and the current parameter set α (see Table 1):

$$R = \left\{ r \mid m \in M \wedge r = \left\| m[p_{img}] - \text{proj}(m[p_{cb}], m[\theta], \alpha) \right\| \right\}. \quad (8)$$

By summing up the squared residuals, the calibration can be formulated as a problem of non-linear least squares:

$$\arg \min_{\alpha} \sum_{r \in R} r^2. \quad (9)$$

To find a parameter set α that minimizes the sum of squared residuals, the algorithm of Levenberg [7] and Marquardt [9] is used, in which the initial parameter set α is set to zero.

Compensation. To compensate for the incorrect servo motor's sensor readings, the optimized joint offsets are added to the requested joint angles and subtracted from the measured joint values. The rotational camera offsets are considered in the calculation of the NAO's camera matrix (see Röfer *et al.* [12]).

3 Results

This section outlines the results of the automatic robot calibration, beginning with a simulated calibration and a concluding calibration experiment executed with a real NAO.

3.1 Simulation

With the help of *SimRobot* [6], the calibration’s feasibility was tested. A simulated NAO was modeled with random erroneous joint offsets and with imprecise checkerboard assembly. As Fig. 4a and b imply, all errors were correctly compensated. Since the errors for each leg chain are different, the residual distribution for the left and right leg exhibit different appearances (see Fig. 4a). Apparent from Fig. 4b, the adjusted parameters create a normal distribution with zero mean and a standard deviation of 0.0486 pixels in x and 0.0403 pixels in y direction.

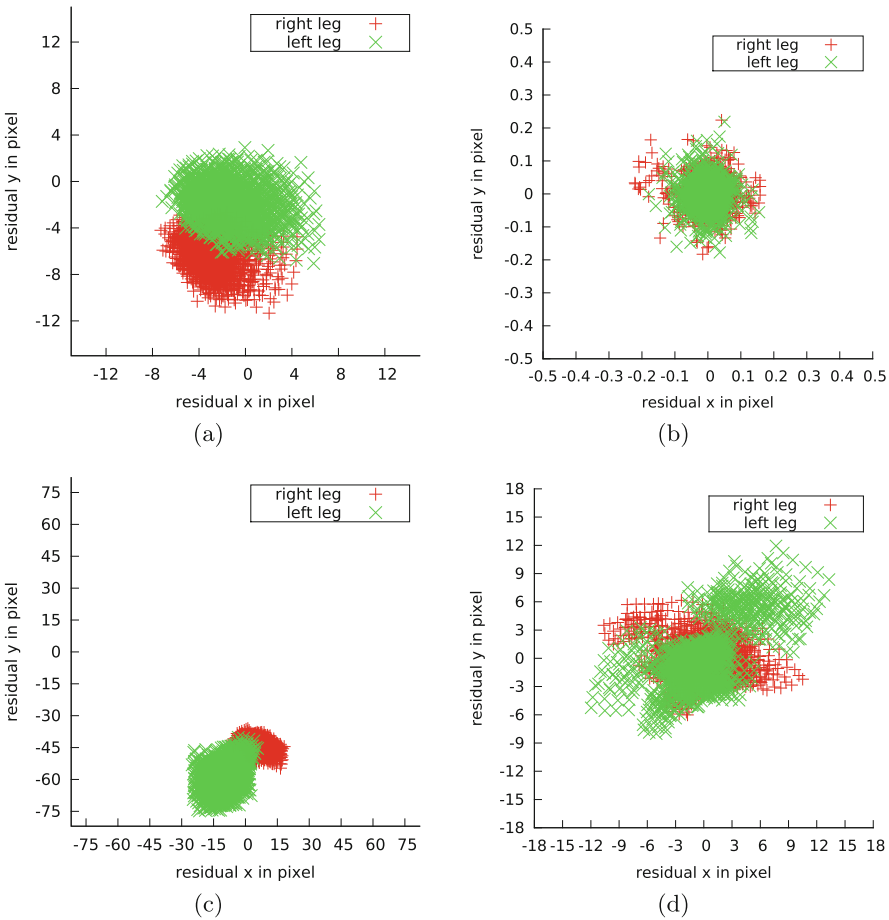


Fig. 4. Residual distribution of a simulated calibration (a) without and (b) with optimized parameters. The lower figures visualize the residual distribution of the first out of three calibration experiments on a real NAO: distribution (c) without and (d) with adjusted parameters. Note that the axes have different ranges.

Table 2. Resulting parameters of three consecutive calibration runs with a NAO.

		1	2	3	\emptyset
α_{CamYaw}	[$^{\circ}$]	0.4764	0.4751	0.4466	0.4660 ± 0.0168
$\alpha_{CamPitch}$	[$^{\circ}$]	-4.5627	-4.5430	-4.4956	-4.5338 ± 0.0345
$\alpha_{CamRoll}$	[$^{\circ}$]	-1.7991	-1.8891	-1.8180	-1.8354 ± 0.0475
$\alpha_{LHipYawPitch}$	[$^{\circ}$]	-2.2938	-2.7094	-2.5055	-2.5029 ± 0.2078
$\alpha_{LHipRoll}$	[$^{\circ}$]	0.9692	0.9620	0.8863	0.9392 ± 0.0459
$\alpha_{LHipPitch}$	[$^{\circ}$]	0.2299	1.3066	0.8783	0.8049 ± 0.5421
$\alpha_{LKneePitch}$	[$^{\circ}$]	-1.1442	-1.9709	-1.6455	-1.5869 ± 0.4165
$\alpha_{LAnklePitch}$	[$^{\circ}$]	1.6708	1.5798	1.6349	1.6285 ± 0.0458
$\alpha_{LAnkleRoll}$	[$^{\circ}$]	0.0267	-0.2078	-0.2618	-0.1476 ± 0.1534
$\alpha_{RHipYawPitch}$	[$^{\circ}$]	-0.9800	-0.9297	-0.9599	-0.9565 ± 0.0253
$\alpha_{RHipRoll}$	[$^{\circ}$]	0.1893	0.3001	0.1881	0.2258 ± 0.0643
$\alpha_{RHipPitch}$	[$^{\circ}$]	1.4916	1.3355	1.3064	1.3778 ± 0.0996
$\alpha_{RKneePitch}$	[$^{\circ}$]	-0.0156	-0.1881	0.1841	-0.0065 ± 0.1863
$\alpha_{RAnklePitch}$	[$^{\circ}$]	0.6081	0.9223	0.6810	0.7371 ± 0.1644
$\alpha_{RAnkleRoll}$	[$^{\circ}$]	-0.5310	-1.0132	-0.7909	-0.7784 ± 0.2413
α_{LRotZ}	[$^{\circ}$]	-0.6010	-0.2357	-0.4890	-0.4419 ± 0.1871
$\alpha_{LTransX}$	[mm]	-0.3878	-1.3591	-0.8144	-0.8538 ± 0.4868
$\alpha_{LTransY}$	[mm]	0.5075	0.7773	0.8117	0.6988 ± 0.1666
α_{RRotZ}	[$^{\circ}$]	0.8058	0.9085	0.7725	0.8289 ± 0.0709
$\alpha_{RTransX}$	[mm]	-2.0173	-1.0657	-1.4765	-1.5198 ± 0.4773
$\alpha_{RTransY}$	[mm]	2.2007	1.7893	1.5274	1.8391 ± 0.3394
<i>Iterations</i>		41	47	35	41
<i>Duration</i>	[s]	591	600	579	590
<i>rms x</i>	[px]	3.22829	3.17493	3.07256	-
<i>rms y</i>	[px]	2.60487	2.53476	2.49331	-

3.2 NAO

On a real NAO, three consecutive automatic calibrations were executed¹. Beforehand, an intrinsic camera calibration was done, resulting in a focal length of 562.5 pixels and an optical center of $(324, 189)^T$ pixels. The resulting parameters, as well as further information, such as the calibration duration, are shown in Table 2. One calibration round took 590s on average and the Levenberg-Marquardt algorithm converged after 41 iterations on average. The root mean squared error was reduced to roughly three pixels in x and two and a half pixels in y direction. It is noticeable, regarding the optimized parameters, that the left

¹ The gathered data of the three experiments can be found as CSV files on: <https://sibylle.informatik.uni-bremen.de/public/calibration/>.



Fig. 5. The checkerboard projection without calibration (a). The resulting projection with optimized parameters α (b).

leg's pitch motor deviations are rather big, with a value of 0.5421° at a max. This implies that there are several different configurations that minimize the sum of squared residuals. In contrast, the right leg's parameters have a maximal deviation of 0.2413° . The vast majority of the parameters were similar after the three calibration runs, in particular the three camera rotation offsets. The checkerboard assembly correction parameters are adequately small.

It is obvious that the resulting parameters are not a perfect solution to the minimization problem, since there are many residuals bigger than the threefold standard deviation (see Fig. 4d). Unlike the depiction of Fig. 5, where (a) is showing the projection of the checkerboard before and (b) after a calibration, there are countless kinematic configurations resulting in an imprecise projection. A standard test for a good calibration of the camera's pose is the projection of the modeled lines of an official SPL field back into the image from a known position on the field and comparing them with the lines that are actually seen. However, using the optimized parameters, this test showed unsatisfactory results. The most probable cause is backlash that has a different impact on a robot lying on its back than on a standing robot.

Using the parameters of the third calibration run, the NAO was able to play soccer for the duration of a half (10 min), while falling down 6 times. Without a calibration, the NAO fell down 11 times in the same amount of time. Further experiments with four other NAOs exhibited a similar result: a calibration never negatively influenced the performance of a NAO playing soccer. In one case, a robot, that was initially not able to walk half a meter, was able to play a complete half (still being very shaky and unstable).

4 Conclusion

In this paper, we presented a method to define an automatic robot calibration as a problem of non-linear least squares, customized for the humanoid robot NAO.

The NAO was modeled with the help of homogeneous coordinate transformation and the pinhole camera model. The resulting least squares problem was solved with the Levenberg-Marquardt algorithm.

The simulated calibrations resulted in perfect estimates for the compensation of the identified errors and therefore prove the plausibility of the presented approach. Performing the calibration with real NAOs turned out to be less satisfying. Apart from the fast overall operation time with roughly 10 min, the deviations of some parameters after consecutive calibrations vary strongly, the projection of field lines was rather imprecise, and there are many joint configurations that resulted in an insufficient re-projection of the checkerboard. However, the kinematic parameters can be used as a better initial guess for a manual calibration.

We think that the extension of the projection model with non-geometric errors, such as joint elasticities, could improve the results. Birbach *et al.* [2] and Wiest [15] modeled elasticities with a spring, using the torques that affect each motor. Since the NAO lacks of torque sensors, only static torques can be used. It is also conceivable that the lengths of the limbs may vary from the values in the official NAO documentation. The most difficult unregarded errors are those that result from backlash, which, according to Gouaillier *et al.* [4], might have a range of $\pm 5^\circ$. In addition, the offsets of the two head joints might also impair the calibration result, because they are not adjusted during the optimization, since they might have a linear dependency to the camera rotation offsets.

The improvement of this calibration approach will be further investigated by considering the possible problems mentioned.

Acknowledgement. We would like to thank the members of the team B-Human for providing the software framework for this work. This work has been partially funded by DFG through SFB/TR 8 “Spatial Cognition”.

References

1. Bennett, S., Lasenby, J.: ChESS - quick and robust detection of chess-board features. *Comput. Vis. Image Underst.* **118**, 197–210 (2014)
2. Birbach, O., Bäuml, B., Frese, U.: Automatic and self-contained calibration of a multi-sensorial humanoid’s upper body. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, pp. 3103–3108 (2012)
3. Elatta, A.Y., Gen, L.P., Zhi, F.L., Daoyuan, Y., Fei, L.: An overview of robot calibration. *IEEE J. Robot. Autom.* **3**, 74–78 (2004)
4. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: The NAO humanoid: a combination of performance and affordability. *CoRR abs/0807.3223* (2008)
5. Kastner, T.: *Automatische Roboterkalibrierung für den humanoiden Roboter NAO*. Diploma thesis, Universität Bremen (2014)
6. Laue, T., Spiess, K., Röfer, T.: SimRobot – a general physical robot simulator and its application in RoboCup. In: Bredendfeld, A., Jacoff, A., Noda, I., Takahashi, Y.

- (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 173–183. Springer, Heidelberg (2006)
7. Levenberg, K.: A method for the solution of certain problems in least squares. *Q. Appl. Math.* **2**, 164–168 (1944)
 8. Markowsky, B.: Semiautomatische Kalibrierung von Naogelenken. Bachelor thesis, Universität Bremen (2011)
 9. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* **11**(2), 431–441 (1963)
 10. Nickels, K.M., Baker, K.: Hand-eye calibration for Robonaut. Technical report, NASA Summer Faculty Fellowship Program Final Report, Johnson Space Center (2003)
 11. Pradeep, V., Konolige, K., Berger, E.: Calibrating a multi-arm multi-sensor robot: a bundle adjustment approach. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) *Experimental Robotics. STAR*, vol. 79, pp. 211–225. Springer, Heidelberg (2012)
 12. Röfer, T., Laue, T., Müller, J., Bartsch, M., Batram, M.J., Böckmann, A., Bösch, M., Kroker, M., Maaß, F., Münder, T., Steinbeck, M., Stolpmann, A., Taddiken, S., Tsogias, A., Wenk, F.: B-Human team report and code release 2013 (2013). <http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf>
 13. Strobl, K.H., Hirzinger, G.: Optimal hand-eye calibration. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pp. 4647–4653. IEEE, Beijing (2006)
 14. Wang, Z., Wu, W., Xu, X.: Auto-recognition and auto-location of the internal corners of planar checkerboard image. In: *International Conference on Intelligent Computing, Heifei, China*, pp. 473–479 (2005)
 15. Wiest, U.: Kinematische Kalibrierung von Industrierobotern. *Berichte aus der Automatisierungstechnik*. Shaker Verlag, Aachen (2001)
 16. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330–1334 (1998)