Providing Ground-truth Data for the Nao Robot Platform

Tim Niemüller[1], Alexander Ferrein[2], Gerhard Eckel[3], David Pirro[3], Patrick Podbregar[4], Tobias Kellner[4], Christof Rath[4], and Gerald Steinbauer[4]

[1] Knowledge-based Systems Group,
RWTH Aachen University, Aachen, Germany
niemueller@kbsg.rwth-aachen.de
[2] Robotics and Agents Research Lab
University of Cape Town, Cape Town, South Africa
alexander.ferrein@uct.ac.za
[3] Institute for Electronic Music and Acoustics
University of Music and Performing Arts, Graz, Austria
eckel@iem.at,pirro@iem.at
[4] Institute for Software Technology
Graz University of Technology, Graz, Austria
{patrick.podbregar,t.kellner.c.rath}@student.tugraz.at,
steinbauer@ist.tugraz.at

Abstract. Collecting ground truth-data for real-world applications is a non-trivial but very important task. In order to evaluate new algorithmic approaches or to benchmark system performance, they are inevitable. This is particularly true for robotics applications. In this paper we present our data collection for the biped humanoid robot Nao. Reflective markers were attached to Nao's body, and the positions and orientation of its body and head were tracked in 6D with an accurate professional vision-based body motion tracking system. While doing so, the data of Nao's internal state, i.e., the readings of all its servos, the inertial measurement unit, the force receptors plus a camera stream of the robot's camera were stored for different, typical robotic soccer scenarios in the context of the RoboCup Standard Platform League. These data will be combined in order to compile an accurate ground-truth data set. We describe how the data were recorded, in which format they are stored, and show the usability of the logged data in some first experiments on the recorded data sets. The data sets will be made publicly available for the RoboCup's Standard Platform League community.

1 Introduction

Collecting ground-truth data for real-world applications is a nontrivial but very important task. In order to evaluate new algorithmic approaches or to benchmark system performance, they are inevitable. This is particularly true for robotics applications. For instance, the European Robotics Research Network (EURON) states in a *Survey and Inventory of Current Efforts in Comparative Robotics*

Research: "It was a well-known fact that the current practice of publishing research results in robotics made it extremely difficult not only to compare results of different approaches, but also to asses the quality of the research presented by the authors." [1]. Therefore, EURON started an initiative on how results in robotics could be better compared, leading to a number of publications and workshops at international robotics conferences. One option for comparing results more easily is to pose a common and non-trivial real-world problem, which is to be solved by a number of research group. Among them are the DARPA Grand Challenge, ELROB, or RoboCup. There is also a number of specialized tasks for establishing benchmarking objectives such as robot manipulating, motion planning, or for mobile robots. As for the latter, the Rawseed project [2] is concentrating on multi-sensor data sets for fundamental problems such as localization and navigation. With such data sets and associated ground-truth data, new localization algorithms can be benchmarked and be compared with existing approaches. Another available robotics data set is, for example, the Radish Data Set [3] or the KTH collection for simultaneous localization and mapping (SLAM) [4][5]. Standardized data sets are also common and available for machine learning approaches and computer vision (e.g. [5–7]).

The purpose of ground-truth data is to provide an absolute reference to a particular set of properties like the position of objects relative to the robot or the robot itself relative to the environment. The ground-truth data has to have a certain accuracy but at least some magnitudes higher than the precision of the algorithm to be evaluated in order to allow for a significant evaluation. Such data sets usually contain a set of sensor data (e.g. laser range scans, inertial measurements, camera images) and a reference position and orientation of the robot. In indoor environment usually a three-dimensional reference (a two-dimensional position plus an orientation) is sufficient. For outdoor environments usually a full six-dimensional reference (a three-dimensional position plus three rotational angles) is necessary.

In the case of a humanoid robot like the Aldebaran Nao [8], which we address in this paper, even more reference data are interesting. In order to perform a task such as playing soccer, the robot has to solve sub-tasks such as object tracking or self-localization. For these sub-tasks, the robot needs an estimation of the position and orientation of its limbs, torso, and head. This information is estimated by processing sensor data of the robot, e.g. inertial measurements, force receptors, joint angles, and camera images. Therefore, it is desirable to have ground truth for the position and orientation of the head and the torso in the reference data set. In order to provide such an extended data set, we conducted experiments in a body motion capturing system with the biped robot platform Nao on a RoboCup Standard Platform League (SPL) football field.

In this paper, we describe the gathering of ground-truth data for the robot. Unlike e.g. [9], where the ground-truth data of the robot were estimated in 3D (x, y, θ) for a localization experiment, we here aim at providing full data sets of the robot's sensory information while performing typical tasks. In particular,

[5] More SLAM related data sets are listed at `http://www.openslam.org`

Fig. 1: The experimental setup for the data recording. The left image shows the RoboCup Standard Platform League field within motion capturing system. The right image shows one of the 15 cameras of the body motion tracking system. The LEDs at the front of the camera provide pulsed infra-red light allowing capturing under day light conditions.

the global 6D coordinates of the robot's head and torso together with the servo values and the camera images offer data sets for also benchmarking algorithms on the Nao platform or conducting off-line experiments, e.g. machine learning.

In the remainder of this paper we describe our experimental setup in Sect. 2. We particularly show the setup of the body motion capturing system and the markers that were attached to the robot during the experiments. In Sect. 3, we briefly go over how the data were logged with our robot control software FAWKES [10], before we show some first experiments on the usefulness of the collected data for benchmarking in Sect. 4. We conclude with Sect. 5.

2 Body Motion Capturing with the Robot Nao

For collecting the extended ground-truth data, we used body motion capturing technology. This technology allows to continuously track the position and orientation of parts of a human body in a three-dimensional space. Moreover, using a model of the human skeleton also the orientation and position of joints can be recorded. The technology is commonly used in movie production, animation, development of computer games, and artistic performances, where the recordings of the natural movements of a human is required.

We used the CUBE laboratory of the Institute for Electronic Music and Acoustics of the University of Music and Performing Arts in Graz. The laboratory is equipped with a high performance motion capturing system and is usually used for research on innovative forms of arts and music [11]. The body motion capturing system is the high resolution fully-3D tracking system V624 by the company Vicon. It consists of 15 infra-red cameras which records the position of a number of reflective markers attached to the body. The system is able to

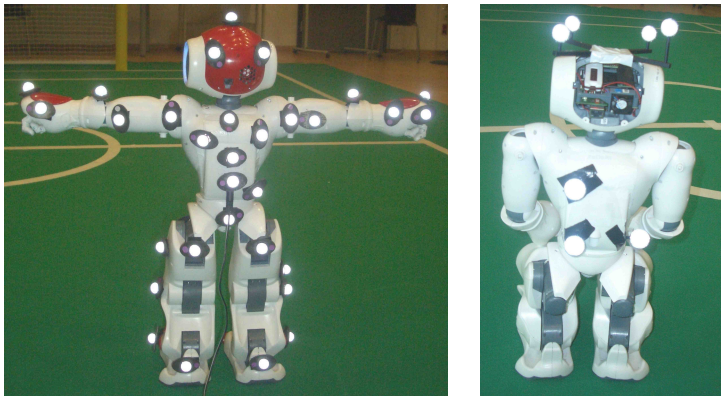


Fig. 2: The experimental setup for the humanoid robot Nao. The left image shows the robot Nao with the 42 tracking markers for a full body motion capturing. The right image shows the robot with four tracking markers each for the head and the torso (the fourth marker on the chest is not visible).

record the positions with a frame-rate of 120 Hz and a high spacial accuracy of less then one millimeter. The system is able to track a volume of about 60 m^3. The left image of Figure 1 shows the experimental setup of the tracking system. In order to be able to record full-featured camera images with the robot, we set up a fully rule-compliant RoboCup Standard Platform League (SPL)[6] field within the body motion capturing system. The right image shows one of the cameras of the tracking system. The camera uses pulsed infra-red light which allows for tracking under day light conditions. This in turn allowed us to record a video stream of the soccer field with the robot's internal camera.

We started with attaching 42 reflective markers to the robot's head, torso, and limbs, just as it would be done for human motion capturing. The left image of Fig. 2 shows the robot with the 42 markers. Unfortunately, it turned out that the robot is too small for a full body motion capturing. Because of the size of the robot, some markers especially on the back and on the feet are too close to each other, which makes it impossible for the system to reliably track these markers. The system fails to distinguish close markers and therefore regularly fails to provide tracking data. Therefore, we decided to track only the position and orientation of the head and torso of the robot. The right image of Fig. 2 shows the robot with markers only attached to the head and torso. We hoped for whole-body data, however, the ground-truth position and orientation of head and torso still provide a reasonable means for benchmarking. The upper body serves as global reference for the robot on the field, while the head provides a global reference for the position and orientation of the cameras which are the main sensors for most applications. The body motion capturing system provides the position and orientation in 6D of two separate rigid bodies modeled as cubes.

[6] SPL website is at `http://www.tzi.de/spl/`

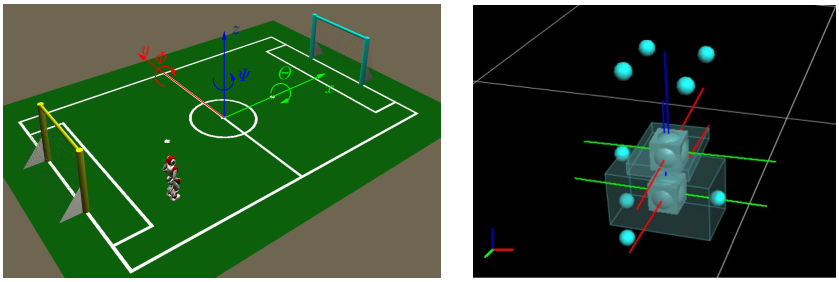


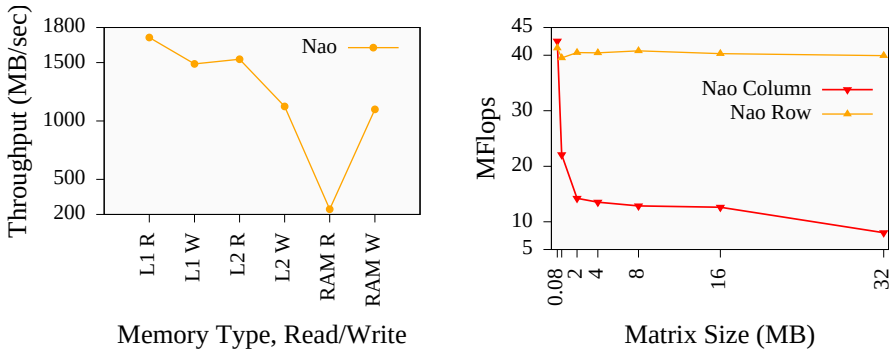
Fig. 3: The coordinate system of the tracking data. The left image shows the global coordinate system in respect to the SPL soccer field. The right image shows the coordinates for the cubes representing the position and orientation of head and torso of the robot with respect to the global coordinate system. The image depicts the alignment of the axis and markers for the robot shown on the left.

The torso has its rotational point in the center, while the head has its rotational point in the head joint. Fig. 3 shows the coordinate system for the ground-truth data.

The left image of Figure 3 shows the 6D coordinate system $(x, y, z, \Theta, \Phi, \Psi)$. The center of this global coordinate system is the center of the field. The x axis runs parallel to the side lines of the field from the yellow goal towards the blue one. The positive y axis runs along the middle line to the left. The z axis points upwards. The right image shows the position and orientation of the two tracking cubes for the head (top) and the torso (bottom). The markers (spheres in the image) are shown in the true alignment with the cubes (refer to Figure 2). The rotation points of the cubes are in the torso's centers and in the head joint, resp. For the torso cube the rotation point is in the center of the upper body behind the chest button. The angles Θ, Φ, Ψ are global and are always related to the global axis. The robot on the left image of Figure 3 has its torso and head in an upright position and is aligned with the global axis. The robot is oriented along the x axis. The connecting line of the shoulder joints is parallel to the y axis. The direction of the rotation of the angles are depict by the corresponding circular arrows.

3 Data Logging

For acquiring useful data sets, it would be desirable to log all the data the robot can provide. However, the throughput of CPU and memory is restricted on the Nao platform, and we therefore had to assess first, how much data we could log. We describe our assessment in Sect. 3.1. In Sect. 3.2, we describe how we logged the data through our robot control software Fawkes.



(a) Memory Throughput (b) CPU Speed

Fig. 4: CPU and Memory Benchmarks for the Nao Robot Platform

3.1 CPU and Memory Assessment

To determine the computational constraints for software on the Nao and the specific problems of logging, we had to investigate the basic capabilities of the robot platforms in terms of CPU speed and memory throughput.

The *memory* of the computing system is divided into several stages, the main memory (RAM) and several levels of cache within the CPU. The AMD Geode LX-800 used in the Nao has two levels of cache, with the L2 cache of 128 KB having more influence on the performance. We have applied a benchmark[7] to test the throughput of the different memory types. The averaged results of 10 runs are shown in Fig. 4(a). The L1 and L2 caches show the expected behavior, with a peak transfer rate of 1.7 GB/sec and 1.5 GB/sec respectively. But the performance of the RAM is much worse, dropping to only 240 MB/sec.

To measure the *CPU speed* an artificial CPU benchmark that implements a matrix-vector multiplication has been used. It implements the calculation of $A = B \cdot C$, where B is a matrix and A and C are vectors. We vary the amount of memory required to store a matrix ranging from 0.08 MB to 32 MB. We have implemented two variants of the multiplication. The first multiplies each row of the matrix B with C, allowing for a sequential access to the matrix memory. The second multiplies each value of a column of B with the appropriate element in C. This requires access to values spread throughout the whole chunk of memory.

Fig. 4(b) shows the results given in million floating point operations per second (MFlops) with double precision. The row-based implementation has an almost constant speed of about 40 MFlops for any matrix size. This indicates that the memory throughput is high enough to keep the CPU busy all of the time as long as the memory can be read optimally. This changes drastically for the column-based implementation. For the smallest matrix the speed is comparable, as the problem of 0.08 MB still fits into the L2 cache of 128 KB. But for larger

[7] Bandwidth benchmark: `http://home.comcast.net/~fbui/bandwidth.html`

matrices the computation speeds drops to 10 MFlops and below. Here we see that the bad throughput from the CPU to the RAM causes the computation units of the CPU to wait for new data most of the time.

3.2 Logging with Fawkes

FAWKES[8] is a component-based robot software framework [10] developed for real-time applications in the field of cognitive mobile robotics. Functionality to operate the robot is encapsulated in plugins which can be combined and configured at run-time.

FAWKES follows a component-based approach for defining different functional blocks. We deploy a blackboard as communication infrastructure. Information is stored as semantic groups of values accessed by a unified interface and identified by a type and a unique ID, but the blackboard also supports message channels between components. The robot data as provided by NaoQi are stored in blackboard interfaces. Data is read from NaoQi at the DCM update rate of 50 Hz[9] and transferred to FAWKES via a remote blackboard connection.

The blackboard is one of the corner stones for the data recording during the experiments. Since all the data produced by the system is stored at a central place it can be logged very efficiently. FAWKES features a generic blackboard logging component. Because all value groups follow a common interface for data access, there is no need for a specific logging mechanism, rather the plugin is only configured for the interfaces which should be logged.

We took the data logs as depicted in Fig. 5. Each log file has a preface describing the data fields contained in the log, which is given as a comma-separated value list. Each line of the ground-truth data consists of a time stamp and the 6D coordinated of head and torso. Each sensor data frame consists of a relative time offset since the start of logging, and the joint angles for all joints. A global time stamp is attached at the end of each frame. Similarly, the electric current values are stored. These data were recorded with a frequency of 50 Hz.

The camera frame were recorded separately. Each frame comes with a global time stamp followed by the camera image itself. The camera image has a resolution of 320×240 in the format YUV 422. The frequency had to be restricted to 15 Hz. With the global time stamps in data and camera frame, both logs can be synchronized.

4 Data Sets

Overall, 20 different scenarios were logged with different setups. Sometimes the robot was running towards an opponent, or the ball, sometimes the robot was just walking around recording camera data. Two example scenarios are shown in Fig. 6. The trajectory presented in these figures are drawn from the tracked

[8] FAWKES is free software and available at `http://www.fawkesrobotics.org`

[9] Experiments were conducted with NaoQi 1.3.17 providing a minimum loop time of 20 ms.

Ground Truth Log

# 6 lines header, describing the text fields													
a	b	c	d	e	f	g	h	i	k	l	m	n	\n

Hardware Log

# 8 lines header, describing the text fields																							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	\n

Hardware Current Log

# 8 lines header, describing the text fields																							
1	2	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	13*	14*	15*	16*	17*	18*	19*	20*	21*	22*	23*	24*
25*	26*	27*	28*	29*	30*	31*	32*	33*	34*	35*	36*	37*	38*	39*	40*	41*	42*	43*	44*	45*	46*	47*	\n

Camera Frame

timestamp	camera frame YUV 422, 320×240@15 Hz
	...
end camera frame	

Legend: [a]time stamp,[b]head roll, [c]head pitch, [d]head yaw, [e]head x, [f]head y, [g]head z, [h]torso roll, [i]torso pitch, [k]torso yaw, [l]torso x [m]torso y [n]torso z, [1]time offset, [2]servo enabled$\in \{true, false\}$, [3]head yaw, [4]head pitch, [5]l shoulder pitch, [6]l shoulder roll, [7]l elbow yaw, [8]l elbow roll, [9]l hip yaw pitch, [10]l hip roll, [11]l hip pitch, [12]l knee pitch, [13]l ankle pitch, [14]l ankle roll, [15]r hip yaw pitch, [16]r hip roll, [17]r hip pitch, [18]r knee pitch, [19]r ankle pitch, [20]r ankle roll, [21]r shoulder pitch, [22]r shoulder roll, [23]r elbow yaw, [24]r elbow roll, [25]acc x, [26]acc y, [27]acc z, [28]gyro x, [29]gyro y, [30]gyro ref, [31]l FSR front l, [32]l FSR front r, [33]l FSR rear l, [34]l FSR rear r, [35]r FSR front l, [36]r FSR front r, [37]r FSR rear l, [38]r FSR rear r, [39]us reading, [40]us direction, [41]l foot bumper l side, [42]l foot bumper r side, [43]r foot bumper l side, [44]r foot bumper r side, [45]chest button state, [46]battery charge, [47]DCM time stamp, note: the starred data yield the current values of the resp. sensor

Fig. 5: Data Frames

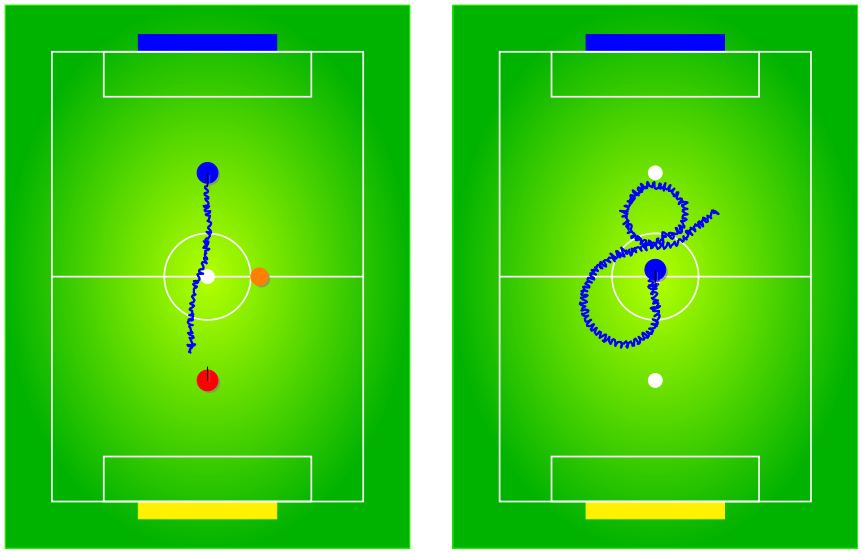


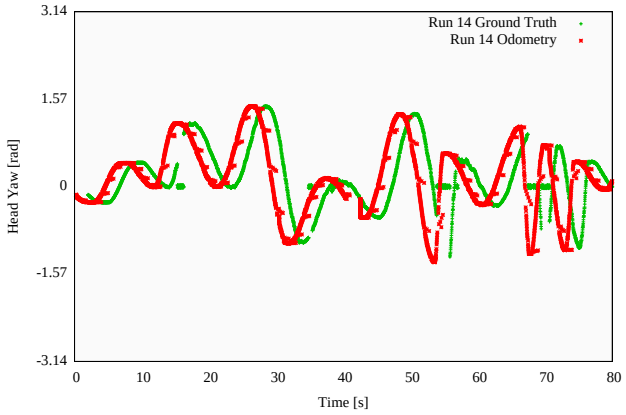
Fig. 6: Example scenarios

(x, y) coordinates of the torso. Note that each trajectory 'wiggles' along the robot's path. The explanation for this is that the torso is tilting to the left and the right, respectively, when the robot walks. As the tracker accuracy is below 1 mm, one can find such movement in the tracked data. The spatial resolution and accuracy of the ground truth data are determined by the motion capturing system. The time synchronization between the data of the tracking system and the data recorded by the robot was done manually. We used easy to find motions like stand-up as synchronization points.

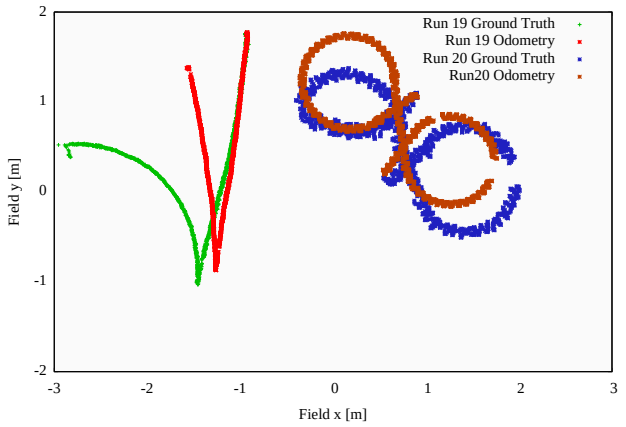
Further, we analyzed the time accuracy of our logging session. In the 20 runs of our current experiments, we could observe some variances in the aspired frame rate depending on the complexity of the task. The data were recorded at frequency of 50 Hz, i.e. 20,000 μsec. The average frame time we could achieve with our robots for the logged data over $92,661$ data frames lies at 21,245.82 μsec, with a standard deviation of 89 μsec.

Fig. 7 shows first experimental results on using the recorded data sets for benchmarking the robot Nao and our software. In particular, we compare the data of the head yaw that the robot logged with the globally tracked data (Fig. 7(a)), the calculated odometry of the robot's position with the tracked relative positions (Fig. 7(b)), as well as the global position, which our localization system yields, with the globally tracked torso position (Fig. 7(c)).

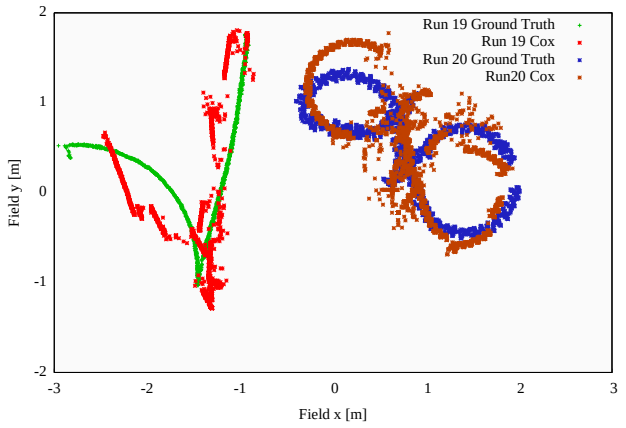
The head yaw data as depicted in Fig. 7(a) were achieved the following way. Several time, the robot was turning around itself (i.e. no translational movement), while his head was constantly moving from the left to the right. This resembles our *search-ball* behavior. The relative head yaw was computed from the tracked data as $head\_yaw_{rel} = normalize(\Theta_{head} - \Theta_{torso})$ with Θ the global



(a) Head yaw odometry vs. ground truth; note that ground truth data were shifted by 1 sec for visibility reasons.



(b) Odometry vs. ground truth



(c) Cox localization vs. ground truth

Fig. 7: Comparison between derived data and tracked data.

yaw angle (see Fig. 3 for the definition of the global coordinate system). The local data were just taken from Nao's head joint reading. Note that the global reference data in Fig. 7 are drawn with a time offset of 1 sec for readability reasons as, otherwise, the graphs would overlay with each other. Also note that the tracker could not always yield data for the torso and the head (gaps in the graph). Moreover, note that the yaw odometry comes from an absolute angle sensor and therefore there the errors do not accumulate like the odometry shown in Fig. 7(b)).

The other graphs show our inverse kinematic calculations of the robot's torso position compared with the globally tracked data, and the results of our localization algorithm (cf. [12] for a concise description of the implementation) for the two data sets *Run-19* and *Run-20*. These two graphs show the usefulness of the recorded data sets, in particular. We now can establish good error measures and compare future enhancements of the kinematic and the localization on these data.

5 Conclusion

In this paper we described how we recorded a ground-truth data collection for the biped humanoid robot Nao. During our experiments we logged all relevant sensor data that the robot Nao can provide, from all sensor joint angles over the joint currents up to the camera stream. An assessment of the throughput of the robot's hardware revealed that the camera stream could not be recorded with a desired 30 Hz at full VGA resolution, but only with 15 Hz at 320×240 pixels. The sensor data were logged with a frequency of 50 Hz, and data analysis showed that this frequency could be satisfyingly achieved in over 90,000 frames. The respective ground truth to our experiments were established by means of a professional motion capturing system that can track human body motion in 6D with a sub-millimeter accuracy. 15 pulsed infra-red cameras were used to track a the head and the torso of the Nao. This technology allowed also for logging the camera images of the robot. In this environment, we set up a football field fully compliant with the RoboCup 2009 SPL rules, and recorded 20 different runs.

As we conducted the experiments very recently, we are still processing the data for compiling the single logs into a coherent data collection. The data sets are available at `http://www.zadeat.org/nao_gt`. Moreover, we are going to develop tool support to provide the data in different formats for allowing real-time playback. By now, we provide the data as comma-separated value text files and in a binary format as it was logged by our robot control software Fawkes. A tool for converting these binary data into comma-separated value files exists already.

In first experiments we showed the plausibility of the data, and that these data sets are meaningful and can be beneficially used for benchmarking both, the robot's hardware as well as the own control software. We compared the robot's head yaw with the globally tracked data and our odometry and pose estimates calculations with the global reference data.

With this paper we hope to start an initiative for benchmarking the platform Nao similar to project like Rawseed by providing reference data sets. This should help for comparing the approaches of the different research groups that work with the Nao.

Acknowledgments

Alexander Ferrein is currently a Feodor Lynen fellow supported by a grant of the Alexander von Humboldt Foundation. This research is also partly supported by the International Bureau of the German Ministry of Education and Research with grant no. SUA 07/031. We would like to thank the anonymous reviewers for their helpful comments.

References

1. Network, E.E.R.S.: Survey and inventory of current efforts in comparative robotics research. http://www.robot.uji.es/EURON/en/index.htm
2. Fontana, G., Matteucci, M., Sorrenti, D.: The RAWSEEDS Proposal for Representation-Independent Benchmarking of SLAM. In: Workshop on Good Experimental Methodologies in Robotics, co-located with Robotics, Science and Systems. (2008)
3. Howard, A., Roy, N.: The Robotics Data Set Repository (Radish) (2003)
4. Folkesson, J.: Kth slam data sets. http://www.nada.kth.se/ johnf/kthdata/-dataset.html
5. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
6. Ponce, J.: Datasets for computer vision research. http://www-cvr.ai.uiuc.edu/ponce_grp/data/
7. Laboratory, L.B.N.: Berkeley image library. http://www.lbl.gov/image-gallery/image-library.html
8. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., 0002, P.L., Marnier, B., Serre, J., Maisonnier, B.: The nao humanoid: a combination of performance and affordability. CoRR **abs/0807.3223** (2008)
9. Laue, T., de Haas, T.J., Burchardt, A., Graf, C., Röfer, T., Härtl, A., Rieskamp, A.: Efficient and reliable sensor models for humanoid soccer robot self-localization. In: Proceedings of the 4th Workshop on Humanoid Soccer Robots (Humanoid 09). (2009)
10. Niemueller, T.: Developing A Behavior Engine for the Fawkes Robot-Control Software and its Adaptation to the Humanoid Platform Nao. Master's thesis, Knowledge-Based Systems Group, RWTH Aachen University (2009)
11. Eckel, G., Pirro, D., Sharma, G.K.: Motion-Enabled Live Electronics. In: Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal (2009)
12. Rath, C.: Self-localization of a biped robot in the RoboCup domain. Master's thesis, Institute for Software Technology, Graz University of Technology (2010)