Dutch Nao Team: Team description for RoboCup 2011 - Istanbul

Visser, A.; ten Velthuis, D.; Wiggers, A.; Verschoor, C.; Cabot, M.; van Egmond, E.; Keune, A.; Fodor, E.; van Zanten, J.; Nugteren, S.; van der Molen, H.; van Bellen, M.; Koster, E.; Laan, S.; Bodewes, B.; Dingeman, T.; Gieske, S.; de Groot, M.; Iepsma, R.; Jozefzoon, O.; Moerbeek, R.; van Rossum, T.; Rozenboom, R.; van Noort, S.

Publication date
2011

Document Version
Author accepted manuscript

Published in
RoboCup 2011: Robot Soccer World Cup XV preproceedings

Dutch Nao Team

Team Description for RoboCup 2011 - Istanbul
`http://www.dutchnaoteam.nl`

Faculty of Science, Universiteit van Amsterdam, The Netherlands

May 23, 2011

1 Introduction

The Dutch Nao Team consists of twenty-four Artificial Intelligence (AI) students, supported by a senior staff-member. The Dutch Nao Team is the continuation of the Dutch Aibo Team; a cooperation of several Dutch universities who were active in the predecessor of the Standard Platform League [9,12,14]. The Dutch Aibo Team has been successful, both in the competition and with a number of publications [3,4,5,7,8,11]. The Dutch Aibo Team always published their source-code online including a technical report about the innovations [6,10,13]. The Dutch Nao Team debuted in the SPL competition at the German Open 2010 [2]. The same year the first paper about research with the Nao was published [1].
As a large team, the responsibilities have been distributed over our team. For the RoboCup the team is divided in a local and remote team. :

Local Supervisor: Dr. Arnoud Visser (assistant professor Intelligent Systems Lab Amsterdam)

Local Team Leader: Duncan ten Velthuis,

Local Captains: Auke Wiggers and Camiel Verschoor

Local team: Michael Cabot, Erik van Egmond, Anna Keune, Eszter Fodor, Justin van Zanten, Sander Nugteren and Hessel van der Molen.

Remote team: Maurits van Bellen, Elise Koster, Steven Laan, Boudewijn Bodewes, Timothy Dingeman, Sharon Gieske, Merel de Groot, Robert Iepsma, Osewa Jozefzoon, Romy Moerbeek, Tim van Rossum , Richard Rozenboom and Sander van Noort

2 Relevant Achievements and Publications

The predecessor of the Dutch Nao Team, the Dutch Aibo Team, was a cooperation of the DECIS Lab, Delft University of Technology, University of Twente, University of Amsterdam and the University of Utrecht. In their first year [13,14], they ported the code of both GT2003 [16] and CMU2003 [17] from the ERS-210 to the ERS-7. They subsequently simplified the behaviour-tree. For the RoboCup 2004 in Lisbon the team concentrated on the Technical Challenges. In the Open Challenge the judges were impressed by their robust sound localization, which earned them a second place. In the second year the code base of the DT2004 and the GT2004 was merged and several innovations were made [10]. For instance, the roles within the team were assigned much smarter and the behaviour of the goalkeeper was redesigned from the very beginning. Which resulted in a very effective goalkeeper. In addition the self-localization was augmented with three innovations, such as a landmark flashback buffer. The result was an improvement of a localization error from 21.3 cm to 14.3 cm. In the final year [6] the code base of DT2005 was further improved. New behaviors were created and existing soccer behaviors were improved. Two innovations were demonstrated: an Automatic Color Calibration [7] was demonstrated on the Dutch Open in Eindhoven and the Visual Compass [3] was demonstrated in Bremen. After the year 2006 the Universiteit van Amsterdam has remained active in the RoboCup community, although in a different league. Since 2004 our university competed in the Rescue League. Since 2008 there is a joint participation with Oxford University, inside the Virtual Rescue competition. In 2010 the Universiteit van Amsterdam bought two academic editions of the Nao and participated for the first time in the Standard Platform League at the German Open [2]. Since the Dutch Nao Team qualified for the RoboCup in Istanbul, the University of Amsterdam is now also in the possession of five v3.3 Nao robots. The Dutch Nao Team participated both at the Rome Open 2011 and in the Iran Open 2011. At the Mediterranean Open Workshop on RoboCup Research, Sharon Gieske and Steven Laan presented "RoboTag with a humanoid robot".

Support

The Universiteit van Amsterdam has been active in the RoboCup since Paris 1998. Once, a team of our university became world champion (UvA Trilearn 2003 in the Simulation League). The cooperation with has been very succesful [1] The Informatics Institute supports the team with a new mobile robot lab (large enough for the Standard Platform League soccer field) and the usage of two academic edition Naos and a package of five v3.3 Naos.

3 Research

As a junior team, the educational aspect is important. The main focus of the University of Amsterdam is the combination of Artificial Intelligence and Robotics. The RoboCup initiative provides the team the opportunity to acquire various abilities of many aspects within robotics. The main goal is to establish a base code build in Python with a focus on motion, vision and localisation. Python is an interpreted, general-purpose high-level programming language whose design philosophy emphasizes code readability. Python aims outo combine "remarkable power with very clear syntax", and its standard library is large and comprehensive. Python is the brainchild of a old student of the University of Amsterdam and was build across the street of our University. Our team will subsequently increase our base codes efficiency and will optimize it through the years to come. And because we believe in the open source community our code will be made available for the RoboCup community.

3.1 Vision

We use the Python OpenCV package to handle our vision modules. OpenCV (Free Open Source Computer Vision) is a library of programming functions mainly aimed at real time computer vision. It has a BSD license (free for commercial or research use). OpenCV was originally written in C but now has a full C++ interface and all new development is in C++. There is also a full Python interface to the library.

[1] see `http://www.jointrescueforces.eu/wiki/tiki-index.php?page=Achievements`.

Object Recognition

OpenCV is used to filter an image for orange pixels. This picture is then blurred to remove possible noise. If a ball is in the original image, the pixel with the maximal 'orangeness is now shown as a clear white spot in the filtered and blurred image. The threshold for a pixel to be part of the ball varies based on lighting. Simple geometry can be used to find the 2D Cartesian location of the ball, as if it were projected on the field. Goals can be detected in a similar way. Filtering an image for the yellow or blue of a goal to find each pole gives us the angle towards the goal. To find the poles, we search through the image from each side and determine which pixels are inliers for the poles. Lines are detected using canny edge and the Hough transform algorithm. Those percepts are used in our Localisation method. It is our intention to use more methods for finding the ball and the goal to have a back-up and if we have several methods, we can switch between them if the first does not find anything. In this way an AnyTime [19] style of algorithm can be made; first use a fast but not that accurate method, followed by a slower and more accurate method. One of these methods is using a colour table so we can look up for every pixel if it's for example part of the field, the ball or just noise. We'd have to make the colour tables at the start of every game to calibrate the colours, because the lightning can be different.

Object Recognition using Feature Analysis

When detecting simple objects, like a ball or a line on the field, it is wise to make use of their simple features. A ball, seen from any direction, is always a circle and a field-line always consists of two parallel lines with similar contrast. Likewise, although they aren't simple objects, the Naos also have noticeable feature: their colored waistbands. All these features can be used to detect the objects on the field. Feature Analysis has our preference over Template Matching for its more dynamic approach. Working with constraints will neglect trivial data such as changes in lighting, color, size or viewing direction. This will probably be used for recognition of other Naos and perhaps even for tracking of the ball. If we are to track Naos, this will be done using EMShift, an algorithm that fuses the meanshift algorithm and feature tracking.

Depth vision

With depth vision the position of an object can be calculated from the Nao's perspective. If the positions of the Naos, the ball, the field-lines and the goals are known, then a 2D top-down representation can be created which maps these objects on the field. Based on this representation an appropriate strategy can be made using a logic programming language, Prolog. The rather fixed environment can be used to our advantage. Sizes of the Naos, goals and the ball seen through the camera can be compared to their real world sizes. Rough estimations of their quantitative positions can be made (a ball of m-pixels across is equivalent to a distance of n-centimeter to the Nao, though this depends largely on environmental circumstances such as lighting) or relative qualitative distances can be calculated (a big Nao is closer than a small Nao).

Predicting External Motion

After mastering object detection, the next step is to make forecasts on what the opponents' Nao's next move will be and where the ball will end up. This will give us the upper hand in making our next move. Based on the current and previous 2D model of the field a prediction can be made on what the next model will look like. Simple assumptions like linear motion can be made about the ball but more advanced techniques are needed to predict the movement of the opponents' Naos. The more previous models are used in these predictions the more accurate these predictions will be.

3.2 Motion

Using the intuitive motion module, designing a motion from scratch is easy. The relatively hard part is making the motions dynamic (alter them based on environment).

3.2.1 Kicking

We have tried experimenting with inverse kinematics but found that maintaining balance and still being able to give the ball a firm kick is quite hard. While this is more accurate than a steady kicking motion, a lot of power is lost. We prefer power over accuracy and have therefore chosen not to make the kick a dynamic one. Only the angle of the kick can be altered, the position of the ball is not taken into consideration. Through experiments and designing using models and the simulator, the maximum angle has been heightened.

3.2.2 Keeping

A dangerous aspect of keeping is the diving motion, since this has the largest chance of damaging the Nao. We try to minimalize possible damage by keeping the Nao as low to the ground as possible before diving, and removing or lowering stiffness before the Nao hits the ground. Landing on the side, without rolling over, also proved quite effective if damage to the head and joints is to be prevented. Sidestepping is a good second option for when the ball moves nearly straight towards the keeper.

Obstacle avoidance

Basing all actions on the 2D model is not enough to assure correct game-play. If the Nao does not have accurate knowledge of its location on the field *obstacle avoidance* must prevent possible cataclysms. Running into the own Naos should be avoided with communication, avoiding opponents' Naos or referees be avoidable with vision or sonar. Although sonar is not very accurate, this should serve as a helpful plan B.

Localization

For localization, a new method will be tested. The idea is to split the field recursively into blocks. Each block holds a probability that a Nao is in this block. Besides a probability, a block contains also a list with the distance (a range) and the angle (also a range) toward each possible observable feature. As features, the goals and line crossings will be used. Children of a block which holds a probability lower than a certain threshold are removed. Is the probability in a block higher than an another threshold, then the block is split. In this way it is possible to maintain a belief over all possible locations without the need to (re)sample. It is assumed that such a tree-like structure results in a method with a lower computational complexity compared to other approaches, while it does hold the possibility to create a complex belief; can estimate the location of the Nao in accurate way; is robust against noisy data; can handle kidnapping; and does not have to keep track of previous observed observations.

Behaviours

An important aspect of soccer in general is the team strategy, the Dutch Nao Team will research several soccer strategies. These strategies will be customized for the RoboCup Standard Platform League playing with four Nao robots. The localization method will be consulted to create team formations suitable for the current game state. Depending on the game state , that includes ball possession and the interim score, offensive or defensive formations are chosen. For each Nao the best fitted role is assigned dependent of the location of all the members of the team. Through communication all Naos will be aware of their roles and the roles of others.

Simulation

A whole new development is the creation of a physical realistic model of an Aldebaran Nao robot inside USARsim [18]. USARsim is a simulation environment based on the Unreal Engine, which enables easily modify the environment around the robot (in contrast with NaoSim) and create realistic lighting condition. In the previous version of USARsim (based on UT2004) a wide variety of robot models was available, including a Sony Qrio and Aibo. The model of the Nao robot is the first type of walking robot in the new version (based on Unreal Development Kit). First a Nao model with two joints (T2) was developed, which was gradually upgraded to a model with 21 joints (T21). The different parts are now nicely scaled, have their corresponding collision frame (see Fig. 1) and are correctly linked.

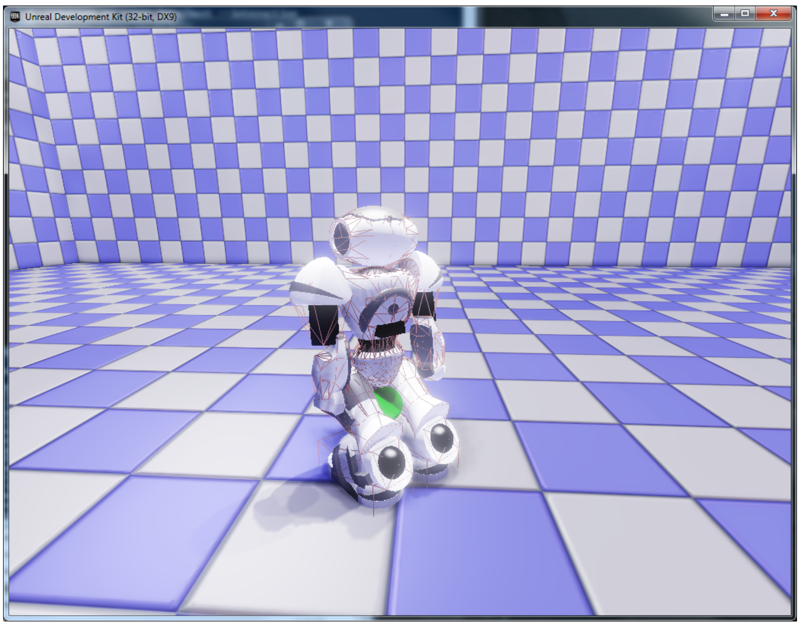

Figure 1: A model of an Aldebaran Nao robot in USARsim, including the collision frame and the center of mass

A remaining task is the calibration of the joints' dynamic behaviors. Important parameters are the mass distribution, the inertia and the gravity. The green circle in Fig. 1 represents the center of mass of all body parts together. With the center of mass close to the hip, representing a mass distribution according to Aldebaran's specifications, the robot currently can stand and keep its balance. Even more complex behaviors as the Tai Chi dance can be executed without loosing balance, although here the dynamics still has to be fine tuned. In the near future the complexity of the tests will be increased, including more complex movements of the Dutch Nao Team.

The development of this open source simulation of a Nao robot not only valuable inside the Standard Platform League, but should also be interesting for the Soccer Simulation League and the @ Home League. Outside the RoboCup community this simulation could be valuable for Human-Robot Interaction research.

4 Conclusion

The Dutch Nao Team has decided to make a clear start and not to base our modules on other teams' implementation of modules. Our main motivation for this is that we want to participate using something original, something built by our team with our team name on it, instead of changing the code of other teams and recycling their accomplishments. Starting from scratch might be hard considering the time constraint but as junior team this can be beneficial in the long run. The Dutch Nao Team tries to distinct (but not distance) itself from other teams, since it is the only Dutch team in the Standard Platform League and is one of the few teams who uses Python.

References

[1] A. van der Mey, F. Smit, K.-J. Droog and A. Visser, ”Emotion Expression of an Affective State Space; a humanoid robot displaying a dynamic emotional state during a soccer game”, Proc. 3rd D-CIS Human Factors Event, November 2010.

[2] A. Visser, R. Iepsma, M. van Bellen, R. Kumar Gupta and B. Khalesi, ”Dutch Nao Team - Team Description Paper - Standard Platform League - German Open 2010”, January 30, 2010 (published online).

[3] J. Sturm and A. Visser, ”An appearance-based visual compass for mobile robots”, Robotics and Autonomous Systems 57 (5), pp. 536-545, 2009.

[4] M. Liem, A. Visser and F.C.A. Groen, ”A Hybrid Algorithm for Tracking and Following People using a Robotic Dog”, in ”Proc. 3rd International Conference on HumanRobot Interaction (HRI 2008)”, pages 185-192, ACM, New York, NY, USA, 2008.

[5] J. Sturm, P. van Rossum and A. Visser, ”Panoramic Localization in the 4-Legged League: Removing the dependence on artificial landmarks” , in ”RoboCup 2006: Robot Soccer World Cup X”, Lecture Notes on Artificial Intelligence series, volume 4434, p. 387-394, Springer, Berlin Heidelberg New York, October 2007.

[6] A. Visser, J. Sturm, P. van Rossum, J. Westra and Th. Bink ”Dutch Aibo Team: Technical Report RoboCup 2006”, December 2006.

[7] D. van Soest, M. de Greef, J. Sturm and A. Visser, ”Autonomous Color Learning in an Artificial Environment”, Proc. 18th Dutch-Belgian Artificial Intelligence Conference, BNAIC'06, October 2006, Namen, Belgium (pp. 299-306).

[8] A. Visser, J. Sturm and F.C.A. Groen, ”Robot companion localization at home and in the office”, Proc. 18th Dutch-Belgian Artificial Intelligence Conference, BNAIC'06, October 2006, Namen, Belgium (pp. 347-354).

[9] A. Visser, P. van Rossum, J. Westra, J. Sturm, D.A. van Soest and M. de Greef, ”Dutch AIBO Team at RoboCup 2006”, Proceedings CD of the 10th RoboCup International Symposium, June 2006, Bremen, Germany.

[10] J. Sturm, A. Visser and N. Wijngaards, "Dutch Aibo Team: Technical Report RoboCup 2005", Technical Report, Dutch Aibo Team, October 2005.

[11] B.A. Slamet, A. Visser, "Purposeful perception by attention-steered robots", Proc. 17th Dutch-Belgian Artificial Intelligence Conference, October 2005, Brussels (pp. 209-215).

[12] N. Wijngaards, F. Dignum, P. Jonker, T. de Ridder, A. Visser, S. Leijnen, J. Sturm, S. Weers, "The Dutch AIBO Team at RoboCup 2005", Proceedings CD of the 9th RoboCup International Symposium, July 2005, Osaka.

[13] S. Oomes, P. Jonker, M. Poel, A. Visser, M. Wiering, W. Caarls, S. Leijnen, S. van Weers, N. Wijngaards and F. Dignum, "The Dutch AIBO Team Report on RoboCup 2004", January 2005.

[14] S. Oomes, P. Jonker, M. Poel, A. Visser, M. Wiering "The Dutch AIBO Team 2004", Proceedings CD of the 8th RoboCup International Symposium, July 2004, Lisbon.

[15] S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", The MIT Press, 2005.

[16] T. Röfer, H.-D. Burkhard, U. Düffert, J. Hoffmann, D. Göhring, M. Jüngel, M. Lötzsch, O. v. Stryk, R. Brunn, M. Kallnik, M. Kunz, S. Petters, M. Risler, M. Stelzer, I. Dahm, M. Wachter, K. Engel, and A. Oster. *GermanTeam RoboCup 2003.* Online, 199 pages, 2003.

[17] M. Veloso, P.E. Rybski, S. Chernova, C. McMillen, J. Fasola, F. vonHundelshausen, D. Vail, A. Trevor, S. Hauert, R. Ros Espinoza. *CMDash05: Team Report.* Online, 12 pages, 2005.

[18] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper USARSim: a robot simulator for research and education, Proceedings of the 2007 IEEE Conference on Robotics and Automation, 2007, pp. 1400-1405.

[19] S. Zilberstei Using Anytime Algorithms in Intelligent Systems, AI Magazine, Vol 17, No 3, Fall 1996, pp 73-83