Dutch Nao Team

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

Faculty of Science, Universiteit van Amsterdam, The Netherlands

December 5, 2010

1 Introduction

The Dutch Nao Team consists of twenty-three second-year 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 as follows over our team:

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

Coordinator: Duncan ten Velthuis

Programmer Captains: Maurits van Bellen, Steven Laan, Auke Wiggers and Justin van Zanten

Programmers: Boudewijn Bodewes, Michael Cabot, Timothy Dingeman, Erik van Egmond, Eszter Fodor, Sharon Gieske, Merel de Groot, Robert Iepsma, Sander Jetten, Osewa Jozefzoon, Anna Keune, Elise Koster, Romy Moerbeek, Remco Mokveld, Sander Nugteren, Tim van Rossum, Richard Rozenboom and Camiel Verschoor

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 and CMU2003 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].

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 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. When qualified, the institute will buy 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.

Initially we will follow the course of the Brown university[1], made with the Robot Operating System (ROS), to get a basic robot playing team. This course is designed for the iCreate robots, which will be ported at the University of Amsterdam to the Nao robot (and published on Nao Share). At the end of the course, when there is a working basis for a soccer team, our team will split up in five separate projects: vision, motion, communication, localization and mapping. The benefit of this arrangement is that every component can be developed independently. Our main goal is to create stable methods for all five components, in order to subsequently increase efficiency and optimize them.

ROS is a framework for robot software development that provides operating system-like functionality on top of a heterogeneous computer cluster. It provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. ROS is based on a graph architecture where processing take place in nodes that may be received, post and multiplex sensor, control, state, planning, actuator and other messages.

ROS has two basic "sides": The operating system side ROS as described above and ROS-pkg, a suite of user contributed packages (organized into sets called stacks) that implement functionality such as localization, mapping, planning, perception, simulation, etc.[2].

ROS offers a programming environment where multiple programming languages like Python, C++ and Prolog can easily communicate with each other. From an artificial intelligent perspective Prolog is a benefit, which is explained further on in this document. The operating system

[1] `http://www.cs.brown.edu/courses/cs148/`

[2] `http://www.ros.org`

offers access to pre-made packages, which help building up a new football team and get the base code going. It also brings access to a large community of knowledge. Like the RoboCup, ROS benefits from their open source community. Therefore, the Dutch code will be made available for the ROS community.

3.1 Vision

Finding Groups

A module is created for finding groups of pixels with the same color. When a snapshot is taken, the picture is converted to a two-dimensional (2D) array. Then the directive is searching the array for a pixel with the given color. If such a pixel is found the search through the 2D array is interrupted and the creation of a group of pixels is started. For every pixel adjacent to the current pixel, it is checked if it is the same color or not. If it is, then the pixel is part of the same group of pixels. For every pixel in the group it is checked if there are adjacent pixels that belong to the group. This process continues until no new pixels are found.

When processing the 2D array, there is the chance that a pixel is found that is already part of a group. Therefore, before starting a new group it is checked if the pixel is not part of any already existing group. What is further done with the found groups depends on the object that is being traced.

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.

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) or relative qualitative distances can be calculated (a big Nao is closer than a small Nao).

Localization

The Nao will localize itself based on the location of landmarks (lines on the field, goals) or on the calculations made by other Naos (read from the shared 2D model). The information will be combined in a Bayesian Filter. MatLab implementations of the EKF, SEIF and FastSLAM algorithms (as described in [15]) are available and will be tested in this setting.

Predicting 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

We have created a module to determine the position of every of the Nao's joints (relative to the Nao's support leg) by presenting the joints respective angles as Euler angles. These Euler angles are used to form a rotation-translation matrix. Then, using simple matrix multiplication, it is possible to calculate from the supporting foot upwards what the relative position of all other joints (and thus body parts) is. From there we should be able to create stable motions and combine these using ROS.

Centroid

From this first module the Nao's centroid can also be derived. The centroid of every part is relative to its respective joint - and the joint's location is now known. For each part of the Nao, it is respective centroid is calculated and these coordinates are used to find the centroid of all parts combined:

$$\vec{r}_{centroid} = \sum \frac{r_i \cdot m_i}{M}$$

Here, r_i stands for the Cartesian coordinates of a specific part i, m_i is the parts mass and M is the mass of all parts (in this case, the mass of the Nao).

Walking

By combining the centroid module, inverse kinematics and a model for the centroid movement during walking, it is possible to create a walking motion. The Nao's movements can also be adjusted dynamically as reaction to unforeseen occurrences in the environment (such as pushing by opponents) detected by our module or the built-in gyroscope, thus preventing it from falling. These movements will be improved with an optimising algorithm. Our main concern is the load on the joints and how to reduce it. The best solution seems to be adjusting the stiffness of the joints during the steps so that the ankle and knee are not overloaded.

Since the ball is not always in the correct position for a kick, the Nao needs to be able to circle around it until the angle towards the goal is minimal, while avoiding pushing opponents. Sidestepping around the ball looks promising. It is a strategy that other RoboCup teams use and usually with success.

Kicks

Last year, the German Open team already developed a more powerful kick [2]. We are planning to make it even better. The centroid module is also used for a kicking module: We will design several kicking motions, each for a different situation. The kicks will be dynamically adjusted since a Nao does not always end up in the perfect situation for a kick. The kicks will be optimized with the genetic algorithm.

3.3 Communication

The Naos are using the ROS communication infrastructure to communicate with each other. One of the advantages of ROS is that it implements several different styles of communication. The buffer transfer is optimized which makes the retrieval of the image faster.

Sharing Knowledge

"Knowledge is power" and the more the Naos know the better. Using ROS, this knowledge can be shared. A Nao will no longer merely rely on the data that it receives via its own sensors. It knows what the rest of the team knows. Accurate measurements made by another Nao can be used to adjust its own measurements. Sharing knowledge will improve the overall accuracy of the 2D representation of the field.

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.

Reasoning

Because of our AI background, our whole team is well known with Prolog, which is supported by ROS. The idea of using Prolog for game analytics means that a logic program can be used to decide our tactics instead of a imperative program, giving our Naos 'real' Artificial Intelligence. Letting the physical actions of the Naos be controlled by other languages (Python, C++) will bring the positive features of a logical language, while negating the negative features, such as slow computation.

4 Collaboration

After a discussion the team stated that collaboration with other universities or teams should be considered as a possibility. The overall agreement is that our team could live with a joined participation, whenever it becomes necessary. There is a lot to learn from cooperation and it will allow both teams to focus on a limited number of research questions.

5 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 itself from other teams, since it is the only Dutch team in the Standard Platform League and is one of the few which uses ROS as operating system in combination with Prolog and 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.