Team Description Paper for RoboCup 2017
Nagoya, Japan

Dutch Nao Team

http://www.dutchnaoteam.nl

Caitlin Lagrand, Sébastien Negrijn, Patrick de Kok, Michiel van der Meer,
Douwe van der Wal, Pieter Kronemeijer, Ozzy Ulger and Arnoud Visser
May 3, 2017

1 Team Information

This is the team description paper for the Dutch Nao Team with Caitlin Lagrand as its team leader. The team consists of two master students, five bachelor students and one staff member from the University of Amsterdam and Maastricht University in The Netherlands. The qualification video is available at our YouTube channel[1]. A research report [1], describing the technical details of the team's work for RoboCup 2016, has previsouly been published on the website[2].

2 Code Usage

Before 2013, the team maintained their own framework in Python. In 2013, the team switched to use Berlin United's code base (then called NaoTH). Because of the lack of documentation, the team decided to use B-Human's framework in 2014 and 2015 for the soccer competitions.

While our previous qualification document mentions the creation of a custom framework based on ROS [2], we later did not enter the competition with this framework as experiments showed that running a ROS [3] core node and image publishing nodelet resulted in a frame rate of approximately 5 Hz even without any further processing. We deemed this too low for usage in RoboCup competitions. In the 2016 RoboCup competitions, we used B-Human's 2015 code release[4].

---

1 https://www.youtube.com/watch?v=FqzGRQzBJwo
2 http://www.dutchnaoteam.nl/publicaties/
3 http://www.ros.org/
4 2015 Release: https://github.com/bhuman/BHumanCodeRelease/tree/1fd87519e2bbb3ccbb5f288980438b692629f7c1

Advancements made by our team include the creation of a new behavior engine and a detection method for the black and white ball which matches candidates by means of regular expressions. A more complete overview can be found in the 2016 technical report[1]. In the recent IranOpen 2017 event a new method was developed to detect the ball, based around low-saturation candidate generation and a deep neural network classification step. This RoboCup we switched, yet again, from the B-Human code base to a new framework, inspired by both the Nothern Bites' RoboGrams architecture [3] and ROS.

2.1 New framework

Our new framework is based on messages sent between modules. A message contains a representation that is provided by a module. Each module can have several update functions to update an output representation using several input representations. The different update functions are connected to each other based on their in- and output representations. The update functions are sorted and executed in order to these in- and output representations.

Apart from sorting based on the dependencies of the update functions, we do not require all representations to be updated every cycle. By doing this we free up more CPU time for some modules, like the ball detector.

2.2 Behavior Engine

Different from the common Finite State Machine approach, the new Behavior Engine of the Dutch Nao Team uses decisions, called axes, which gradually gain score depending on environmental factors. The environmental factors that an axis is based on can be any method that is available in the system. Various frequently used methods are wrapped into a small library, simplifying the search for suitable functions[5]. Each axis thus depends on a number of environmental factors, we call these factors considerations. Considerations are paired into a graph structure, the root of the graph determines the value of the scoring axis. The final score of a consideration could be a non-linear combination of scores returned by the environmental functions and enables us to describe complex relations intuitively. These environmental functions are collectively called the utility function $u(d)$ of the consideration. All consideration scores $C(d)$ are multiplied and normalized into score $s(a)$ and the action of the decision d with the highest score, chosen with Equation 1, is executed.

$$d = \arg\max_{d \in D} u(d) \prod_{a \in C(d)} \sigma(a) \tag{1}$$

This new Behavior Engine has proven to be a versatile tool to create new behaviors, including specifying under which specific circumstances they should be applied.

2.3 Ball detection

In the past the team's ball detection was based around regular expressions on pixel columns, also known as scan lines, converted to a string based on the color calibration. This did not work well because there were too many false positives and this method relied too much on color calibration, which will get even more problematic with realistic lighting.

[5] Publicly available at: https://github.com/pkok/behavior-engine

The new ball detector is based around a candidate generator and a convolutional deep neural network to classify the candidates.

When the lighting changes during a game, the perceived colors change as well. Therefore the ball detector is designed to be as much color-independent as possible. Fortunately, one aspect of vision stays about the same in different lighting conditions: the saturation of pixels. Both black and white will be guaranteed to have a low saturation. Scanlines are used to not process the entire image. If there is a short line of low saturation pixels, this area will be processed further. A square is placed around the line with low saturation in the image and based on an expected ball size, a decision is made whether there are enough pixels with a low saturation to contain a ball. If so, the area is passed to the classifier, which will return a value between 0 and 1, indicating how confident the network is that the candidate is a ball. The only calibrations that are still needed with this setup are a threshold for low saturation and a confidence threshold for accepting a region of the image as a ball.

The convolutional neural network is made using Tiny DNN[6], every candidate is scaled to a size of 18*18*3 and processed through 4 convolutional layers. The network has been trained on several thousands of images, but needs some more training on the edge cases, such as false positives in a robot's own body.

2.4 Deep Neural nets

Based on the recent enhancements in the speed up of (convolutional) neural networks, the Dutch Nao Team is working to replace the current vision methods by a single neural network. Similar to other research in the field of object segmentation [4] this network outputs a probability for each pixel in the image of belonging to an object class (ball, goal, field, robot, etc.). As typical networks proposed in these applications are very deep, we combine the results from recent papers [5] that provide significant speed up by only using binary values in most layers solving the limitations of the Nao robot.

2.5 Localization

The localization method currently being used is replaced with a method that detects points at line segments in the image and match those points with the classic Iterative Closest Point algorithm [6] to the known dimensions of the field. First the internal sensors of the Nao are used to predict the next pose of the robot (motion model), followed by a sensor update based on the estimate of the Iterative Closest Point algorithm (sensor model). Both the pose estimates of the motion model and the sensor model are combined on the confidence in both estimates, resulting in an implementation of an Extended Kalman Filter for localization [7].

2.6 Team Communication

As of now, the robots from the Dutch Nao Team do not share information about their intentions and perception except for the drop-in challenge. Similar to how the decision engine works for a single robot, a same approach is being developed to determine a team strategy using shared information. The shared information is modeled with an uncertainty after which the best role (defender, striker, etc.) is assigned to each robot.

[6] https://github.com/tiny-dnn/tiny-dnn

3 Impact

In the past two years the Dutch Nao Team has provided its support or resources in four publications and projects that lead to a publication ranging from a large variety of topics. Using the Optitrack[7] camera tracking system, which is usually used to track our Nao robots for ground truth localization information, a worm from the Open Worm project [8] was followed with an aerial drone [9]. The Dutch Nao Team also extends its applications of the Nao robot to the @Home league of the RoboCup: in another project the Nao robot was used to help in a kitchen environment by finding a tomato and grabbing it from a table [10]. Finally, the Dutch Nao Team has made the penalty shootout situation into a standalone demonstration [11] which it premiered at the Benelux Conference on Artificial Intelligence and won the first prize for best demonstration.

Earlier the Dutch Nao Team has published papers in the International Conference on Advanced Robotics [12], the Performance Metrics for Intelligent Systems Workshop [13], the RoboCup IranOpen Symposium [14], the RoboCup Symposium [15] and the international conferences as International Conference on Autonomous Robot Systems and Competitions [10].

The Dutch Nao Team is not the only team of the Intelligent Robotics Lab; students and experience are shared with teams participating in the RoCKIn@Work camp [16], the HumaBot competition [17] and the RoboCup@Rescue [18].

4 Other

The Intelligent Robotics Lab has submitted a proposal to compete in the standard platform competition of the @Home league with a SoftBank Robotics Pepper robot [19]. Two former members of the Dutch Nao Team will focus their research on Pepper. Both teams will work closely together, exchanging their insights and experiences on humanoid robots.

In addition, the Dutch Nao Team has been active to promote robotics research to a broad audience, for instance with a performance on the stage of the National Theatre at the Science Gala.

5 Conclusion

The Dutch Nao Team has developed several new techniques to advance their soccer skills. The RoboCup is the perfect platform for testing these techniques. The team has established a foundation dedicated to the team to make crowd funding possible. With this sort of sponsoring and support it will be possible to compete in several RoboCup competitions no matter the location.

References

1. Lagrand, C., van der Meer, M., Gerbscheid, J., Groot, T., Negrijn, S., de Kok, P.: Dutch nao team - technical report. Technical report, Universiteit van Amsterdam, FNWI (2016)
2. de Kok, P., Negrijn, S., Karaalioğlu, M., Lagrand, C., van der Meer, M., Gerbscheid, J., Groot, T., Visser, A.: Dutch Nao Team Team Qualification Document for RoboCup 2016 - Leipzig, Germany (November 2014)
3. Mamantov, E., Silver, W., Dawson, W., Chown, E. In: RoboGrams: A Lightweight Message Passing Architecture for RoboCup Soccer. Springer International Publishing, Cham (2015) 306–317

[7] http://optitrack.com/

4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
5. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279 (2016)
6. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Robotics-DL tentative, International Society for Optics and Photonics (1992) 586–606
7. Gutmann, J.S., Fox, D.: An experimental comparison of localization methods continued. In: Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on. Volume 1., IEEE (2002) 454–459
8. Palyanov, A., Khayrulin, S., Larson, S.D., Dibert, A.: Towards a virtual c. elegans: A framework for simulation and visualization of the neuromuscular system in a 3d physical environment. In silico biology **11**(3, 4) (2012) 137–147
9. van Harmelen, M., van der Meer, M., Boon, M., Gerbscheid, J., Visser, A.: Hunting a robot controlled by an artificial brain. In: Proceedings of the 27th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC 2015), Hasselt, Belgium (2015)
10. Lagrand, C., van der Meer, M., Visser, A.: The roasted tomato challenge for a humanoid robot. In: Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Bragança, Portugal (2016)
11. Lagrand, C., de Kok, P., Negrijn, S., van der Meer, M., Visser, A.: Autonomous robot soccer matches. In: BNAIC2016 Proceedings. (2016) 237–238
12. Wiggers, A., Visser, A.: Discovering reoccurring motifs to predict opponent behavior. In: Proceedings of the 16th International Conference on Advanced Robotics, Montevideo, Uruguay (2013)
13. van Noort, S., Visser, A.: Validation of the dynamics of an humanoid robot in usarsim. In: Proceedings of the Performance Metrics for Intelligent Systems Workshop (PerMIS'12). (2012)
14. Visser, A., de Bos, D., van der Molen, H.: An experimental comparison of mapping methods, the gutmann dataset. In: Proceedings of the RoboCup IranOpen 2011 Symposium (RIOS11). (2011)
15. van Noort, S., Visser, A.: Extending virtual robots towards robocup soccer simulation and @home. In: RoboCup 2012: Robot Soccer World Cup XVI. Springer (2013) 332–343
16. Negrijn, S.: RoCKIn @ Work Visual Servoing – active vision using template matching on rgb-d sensor images. Bachelor's thesis, Universiteit van Amsterdam (June 2015)
17. Cervera, E., Garcia, J.C., Sanz, P.J.: Toward the robot butler: The humabot challenge [competitions]. Robotics & Automation Magazine, IEEE **22**(2) (2015) 8–17
18. Karaalioğlu, M.: Real-time 3-d weighted scan matching with octrees. Bachelor's thesis, Universiteit van Amsterdam (August 2015)
19. Gerbscheid, J., Groot, T., Visser, A.: UvA@Home 2017 standard platform proposal. University of Amsterdam (November 2016)