

The NUbots Software Overview for RoboCup 2020

Matthew Amos, Alex Biddulph, Stephan Chalup, Daniel Ginn, Alexandre Mendes, Timothy Mullen, Josephus Paye, Ysobel Sims, Anita Sugo, Peter Turner, and Taylor Young

Newcastle Robotics Laboratory
School of Electrical Engineering and Computing
Faculty of Engineering and Built Environment
The University of Newcastle, Callaghan 2308, Australia
Contact: nubots@newcastle.edu.au
Homepage: <http://robots.newcastle.edu.au>

1 Walking

The current walk implementation uses an open-loop hybrid walk controller, dynamically switching between an analytically solved trajectory for the centre of mass based on the linear inverted pendulum model, and a Zero Moment Point preview controller, based on the work of Yi et al. [11] and Song [10]. To add local feedback, positions for the pitch and roll ankle joints are prescribed based on information from the Inertial Measurement Unit (IMU) to counteract small imbalances.

Our current walk is an old walk used on the Darwin-OP robots and has major issues. Because of this we are investigating other walks to use for the 2020 competition. These include BitBot's Quintic Walk [4] based off Rhoban's Quintic Walk [8] and Rhoban's IK Walk [8]. A quasi-static walk being developed as a mechatronics final year project is due to be completed by the 2020 competition. It uses a non-linear optimisation method to generate a quasi-static walk gait. A separate project involves developing a static walk using a dynamic foot and torso path planner.

2 Vision

The Visual Mesh [5] underpins the vision system, and is used for sparse detection of balls, points on the field, field lines, goal posts and other robots.

The Visual Mesh is a convolutional neural network for object detection based on a mesh which samples pixels from an image. The mesh is depth independent and the number of mesh points is small enough to allow a CNN to run in real-time on a humanoid robot. Tests on the CPU from the Intel[®] NUC7i7BNH show that a nine layer visual mesh had an execution time of 2.44ms [5]. Furthermore, the visual mesh does not degrade in accuracy when objects occur at different

distances. This is due to the depth independence introduced by using the mesh. The visual mesh was used to detect a soccer ball on a field up to 10m away from the camera [5]. The visual mesh can detect soccer balls, field lines and goal posts, however it is built assuming the object to be detected is spherical.

The visual mesh is created using the following geometric assumptions for each image:

1. The camera lens type, height and orientation with respect to the ground are known.
2. The object is spherical with known radius and remains on the ground.

The height, orientation and radius are used to create a set of unit vectors based at the camera position. Any vector with origin at the camera can be thought of as a ray of light travelling towards the camera. If the vector is within the field of view of the camera it will be collected by the lens. The vector is mapped to a point on the image using the lens projection equation, provided the point is within the bounds of the image sensor of the camera. The visual mesh is formed by taking an array of vectors and associating each of them with a point and pixel in the image. The array of vectors is specially constructed using two equations to efficiently sample the space around the camera for the object. This means the points (sample points) on the image will efficiently sample the image for the object. Because the vectors are generated in the space around the camera, the sample is consistent despite changes in lens type, image resolution, and the apparent size of an object due to distance. Each sample point is connected to its six closest neighbours and these connections become the edges of the mesh. A fully convolutional neural network [9] is used on the mesh where each sample point and it's neighbours become the input to a convolution. All layers in the network use seven points convolutions since there are six neighbours for each sample point. A parameter controls the number of sample points on the object which determines the level of detail available to the network.

The height and orientation of the camera is tracked using the kinematics and inertial measurement unit of the robot. The radius of the soccer ball is given. The unit vectors only need to be calculated once for each height and radius pair. As the robot walks its height varies. A series of visual meshes for different heights are calculated on startup. For each image this information is recorded and the appropriate visual mesh is chosen. A binary search pattern is used to select the vectors that will become points on the image based on the camera's orientation. The partitions of the binary search pattern are built on startup and the search is run in real-time. Every training and run-time image is converted into the mesh before the network is run. The network labels each point of the mesh with the probability it belongs to an object class. A post-processing algorithm based on the probabilities of the mesh can cluster the sample points to determine an object's position.

From the Visual Mesh a series of specialised detectors are employed to detect field edges, balls, and goal posts. Firstly, all points that the Visual Mesh has identified as either field points or field line points are clustered into connected regions and, each cluster is then either merged or discarded using some heuristics

until a single cluster remains. Finally, an upper convex hull algorithm is applied to the final cluster determine the edge of the field.

The ball detector forms clusters out of all the points that the Visual Mesh has identified as ball points. Specifically, the clusters are formed from Visual Mesh points that are identified as being a ball point, but have at least 1 neighbour which is *not* a ball point. This allows us to form clusters of ball edge points. Any clusters which are not below the field edge are discarded. A circular cone is then fitted to each cluster. The cone axis is determined from the line segment between the centre on the camera and the average of all ball edge points. The radius of the cone is determined by the maximum distance between the ball edge points and the average of all of the ball edge points. Different heuristics, such as degree of circle fit and different distance metrics, are then used to discard to cones.

The goal post detector follows a similar structure to the ball detector. Clusters are formed from goal post edge points and any clusters that do not intersect the field edge are discarded. The bottom centre point of the goal post is then found by averaging the edge points. The distance to the goal posts are determined and if there are multiple goal posts detected an attempt is made to assign *leftness* and *rightness* to each post.

At present, a field line detector is not implemented, but will likely be implemented before the next competition.

All calculations in all detectors are done in 3D world coordinates.

3 Localisation

The localisation on-board the robot is performed using a Particle Filter. This allows us to maintain multiple hypotheses about the current pose of the robot, providing robustness against the mirrored field problem and having multiple potential initial positions when entering the field. The filter relies on measurement updates from the vision module, and on IMU data for time updates. The measurement update previously only tracked the four goal post locations; however, we are working on introducing the tracking of field lines during 2020. In particular, we will be focusing on distinctive field features such as corners and the centre circle.

4 Behaviour

The robot's behaviour is a basic state machine where the robot will look for the ball and goals, position itself to kick the ball toward the goals, and then kick. The vision system determines which foot the robot will kick with.

5 Contributions to RoboCup

The NUbots team participated in the 2019 Humanoid Teen-Size League and finished as quarter finalists. The NUbots have participated in the Four Legged League (2002-2007), the Standard Platform League (2008-2011), the Kid-Size Humanoid League (2012-2017), and the Teen-Size Humanoid league (2018-2019). NUbots were the Four Legged League world champions in 2006. The team won the first Standard Platform League in 2008 as team NUManoid in collaboration with the National University of Maynooth, Ireland.

The team has all its RoboCup code [6], hardware [12], and debugging tools [2] developed as open source on GitHub.

The NUbots team have developed a Blender plugin to generate semi-synthetic images with fully-annotated ground truth segmentation maps [1]. The images contain random ball positions, robot positions and kinematic poses, obstacles, and viewer orientations. This tool is public on GitHub for anyone in the League to use.

The NUbots team have been active participants in the Humanoid League rules discussions. A new standard communication protocol [3], based on Protobuf messages, was proposed by NUbots to the TC. A prototype tool, based on the NUsight [2] debugging utility, for monitoring network communications and displaying robot communications in a meaningful manner is currently being developed.

A new development this year hopes to benefit not only the NUbots team but other teams interested in any of our systems. This is a comprehensive documentation resource in the form of a public website [7], providing detailed information about the hardware and software systems, as well as current and future projects.

References

1. M. Amos and A. Biddulph. NUbots PBR pipeline repository. <https://github.com/nubots/NUpbr>.
2. B. Annable, T. Houliston, M. Olejniczak, J. Paye, and A. Biddulph. NUsight2 real-time web-based debugging utility code repository. <https://github.com/nubots/NUsight2>.
3. A. Biddulph. NUbots robocup communication protocol repository. <https://github.com/NUbots/RobocupProtocol>.
4. Team BitBots. Bitbot's quintic walk. https://github.com/bit-bots/bitbots_motion/.
5. Trent Houliston and Stephan K. Chalup. Visual mesh: Real-time object detection using constant sample density. *CoRR*, abs/1807.08405, 2018.
6. M. Metcalfe, J. Fountain, A. Sugo, T. Houliston, A. Biddulph, A. Dabson, T. Johnson, J. Johnson, B. Annable, S. Nicklin, S. Fenn, D. Budden, J. Walker, J. Reitveld, Y. Sims, T. Young, M. Amos, and D. Ginn. NUbots robocup code repository. <https://github.com/nubots/NUbots>.
7. J. Paye and Y. Sims. NUbots team handbook. <https://github.com/NUbots/NUbook>.
8. Team Rhoban. Rhoban's quintic walk. <https://github.com/Rhoban/model/>.

9. E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017.
10. Seungmoon Song. Development of an omni-directional gait generator and a stabilization feedback controller for humanoid robots. 2010.
11. S. Yi, D. Hong, and D. D. Lee. A hybrid walk controller for resource-constrained humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 88–93, Oct 2013.
12. T. Young and A. Biddulph. NUbots hardware repository. <https://github.com/nubots/NUware>.