Nao recognition and coordination

Sander Nugteren

August 24, 2012

1 Project

In RoboCup soccer, it is important to have robots communicate and position themselves properly. This prevents them from getting in the way of other players and enables them to position themselves more favourably. Since this year, this has become even more crucial, as the Standard Platform League (SPL) games now take place on a field with two goals of the same color. This necessitates cooperation between the robots to localise themselves. The robots used in SPL, the Aldebaran Nao robots, have wifi to send and recieve messages.

Figure 1: A Nao by Aldebaran Robotics

The aim of this project is to get our Nao robots to walk in formation. This has already been done by UT Austin Villa in the 3D Simulation League

([2]), which also uses simulated Nao's. The project's aim will be to see how this research could be applied to the real world.

To get the robots to walk in formation, they need the ability to localize themselves, at least with respect to eachother. To do this, they need to be able to recognize other robots. Detection of other robots can also be useful for obstacle avoidance. Initially this would be done with AprilTag ([3]), a tag recognition system developed for use with robotics. However, due to compatibility problems with the existing Dutch Nao Team code base, and the fact that the use of these tags might be in conflict with the game rules (greatly reducing the usefulness of this work), it was decided that an approach using the waistbands and slightly modified numbers would work better.

The first section will be about the coordination algorithm, the second section will outline the Nao recognition algorithm, the third chapter will be a short summary of the results, and the final chapter will be a short conclusion and the future work that needs to be done.

2 Coordination

The coordination problem is trying to give the optimal position assignments to a group of robots given their positions. Here the total distance that each robot has to travel has to be minimized. The most straightforward way of doing this would be to try all mappings, but this would have a complexity of n! for n positions. However, the solution as outlined in the paper by Austin Villa ([2]) solves this problem using one of its properties: All subsets of an optimal mapping between robots and positions are themselves optimal.

The algorithm works by taking a robot position and computing the distance to all goal positions. Then, for each of these the optimal map for the rest of the robots is computed by recursively using the same role assignment function. Then the cost of the role assignment is added to the distance between the robot and goal position computed earlier. The mapping which minimises this is chosen as the best one.

Algorithm 1 Dynamic Programming Implementation. Courtesy Patrick MacAlpine, Francisco Barrera, and Peter Stone

Hashmap $BestRoleMap = \emptyset$
$Agents = \{a_1, ..., a_n\}$
$Positions = \{p_1, ..., p_n\}$
for k = 1 to n **do**
  for each a in $Agents$ **do**
    $S = \begin{pmatrix} n-1 \\ k-1 \end{pmatrix}$ sets of $k-1$ agents from $Agents - \{a\}$
    for each s in S **do**
      Mapping m_0 = bestRoleMap[s]
      Mapping m = $(a \to p_k) \cup m_0$
      bestRoleMap[$\{a\} \cup s$] = mincost(m,bestRoleMap[$\{a\} \cup s$]
    end for
  end for
end for
return bestRoleMap[$Agents$]

3 Nao Recognition

As mentioned before, Nao recognition is an important aspect of this project. This can be done by searching for the waistbands the robots wear to differentiate the teams. The current method just searches for the color of a waistband (red or blue) and uses the information of the location and angles of the head relative to the middle of the feet. The typical height of the waistband of a Nao standing up is also used to compute the distance to the Nao.

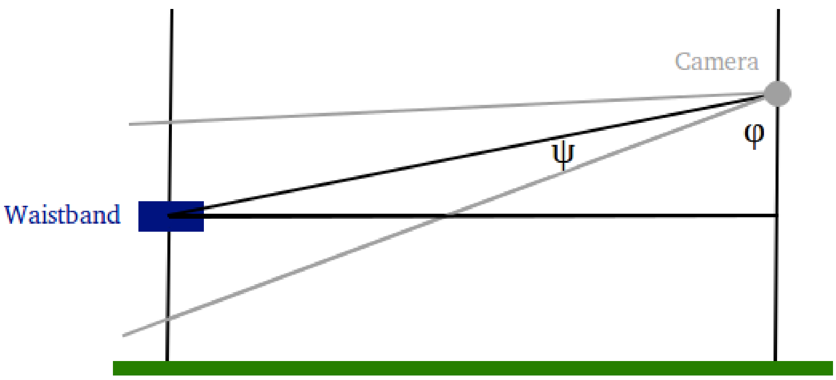

Figure 2: Sketch of the geometry involved in estimating the distance of another Nao. The camera viewing range is given by the grey lines.

To find the distance to the waistband, the angle of $\phi+\psi$ has to be found.

The angle of ϕ is known as this is the angle of the camera found by using forward kinematics. ψ is given by multiplying the pixel coordinate by the total viewing angle of the camera, and then divide it by the amount of pixels in that dimension: $\psi = coord \times \frac{cameraAngle}{resolution}$ The distance to the waistband is given by $\tan(\phi + \psi) \times (cameraHeight - waistbandHeight)$. This yields the distance in the x direction. The angle in the y direction is obtained in a similar fashion. The distance to the waistband in the y direction is given by $\tan(yAngle) \times xDistance$.

In other papers on Nao recognition a similar approach was used (for example, [1] and [4]), but these approaches also checked for the white color around the waistband. The current implementation does not do this, but this should be less of a problem as it used to be, because the main sources of false positives were the blue goal posts, which have been abandoned this year. If the color ranges are set correctly, the amount of other false positives should be reduced to a minimum.

4 Results

To test the accuracy of the Nao detection algorithm, a Nao with a blue waistband was placed approximately 80 cm in front of another Nao. Both robots were stationary. The benchmark of the speed of the algorithm was done by counting the iterations of the algorithm over 10 seconds on a Nao V4. This resulted in a framerate of approximately 23 frames per second.

The accuracy of the estimate was also tested in this setup. The position of the Nao relative to the observing Nao was estimated with an accuracy of about 5 centimeters in both the x and y direction. This already makes this method precise enough to use for obstacle avoidance. The coordination algorithm is also very efficient. A benchmark with simulated data of three robots got to about 15000 iterations per second on the same Nao V4.

5 Future Work

To truly combine Nao recognition and coordination, not only the position of each Nao must be known, but also the heading. In addition to that it must be known which Nao is seen. For this a simple numbering system was used, with the amount of vertical lines representing the number, and the number of horizontal lines the position on the Nao (front, back, left or right). However, the algorithm to recognise these numbers was not ready at the time of making this report, so this is why it is not mentioned in the report itself.

After this, an algorithm is needed to map the measurements of an individual robot to a shared world state. This also brings some problems as the graph respresenting which robot sees which could (at least in theory) contain cyclical relations (robot A sees robot B, robot B sees robot A). This could maybe be solved with a Simultaneous Localization And Mapping (SLAM) algorithm but a computationally lighter method may yield acceptable results as well.

The Nao recognition algorithm can also be improved. The current algorithm searches for the Region Of Interest (ROI) in the image, and takes this as the position of the waistband. This is fine for detecting one Nao, but if there are multiple Naos in view only the Nao with the best visible band will be seen. It would be better to search for blue or red rectangles in the image, because then fallen robots can also be detected correctly (as then the height of the waistband will be higher than the width).

6 Acknowledgements

The author would like to thank Arnoud Visser for supervising this project, as well as the Dutch Nao Team for providing the software base for this project.

References

[1] Alexander Fabisch, Tim Laue, and Thomas Röfer. Robot recognition and modeling in the robocup standard platform league. In Changiu Zhou, Enrico Pagello, Sven Behnke, Emanuele Menegatti, Thomas Röfer, and Peter Stone, editors, *Proceedings of the Fourth Workshop on Humanoid Soccer Robots*. o.A., 2010.

[2] Patrick MacAlpine, Francisco Barrera, and Peter Stone. Positioning to win: A dynamic role assignment and formation positioning system. In Xiaoping Chen, Peter Stone, Luis Enrique Sucar, and Tijn Van der Zant, editors, *RoboCup-2012: Robot Soccer World Cup XVI*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2013.

[3] Edwin Olson. Apriltag: A robust and flexible multi-purpose fiducial system. Technical report, University of Michigan APRIL Laboratory, May 2010.

[4] Maurits van Bellen, Robert Iepsma, Merel de Groot, and Arjen Swellengrebel. Zone blocking. Technical report, Universiteit van Amsterdam, 2011.