wrighteagle unleashed!

RoboCup 2010 Team Report

Standard Platform League

University of Technology, Sydney

James Hornitzky, Benjamin Johnston, Alankar Karol, Rony Novianto, Gavin Spargo, Wei Wang, Xun Wang, Glenn Wightwick (IBM), Mary-Anne Williams

University of Science and Technology of China

Dongpeng Chen, Xiaoping Chen, Min Cheng, Yu Gong, Jinsu Liu, Hao Sun, Ming Zhong

Version 1.0; 5 December 2010

Introduction

The University of Science and Technology of China (Wright Eagle) and the University of Technology, Sydney (UTS Unleashed!) have had a long-term involvement in several leagues at RoboCup including the Standard Platform League. In 2008 we experimented with a joint team called WrightEagleUnleashed! using the Sony AIBO robots and were Runners Up at RoboCup 2008. In 2009 USTC teamed up with CMU and reached the Quarterfinals. Towards the end of 2009 USTC and UTS started to put together a completely new joint team to compete in the Nao SPL in 2010 and 2011 and set about gathering interested students, researchers and resources.

International research collaboration adds to the underlying challenge of a joint team across Australia and China must overcome major barriers of language, culture, distance and time-zones in addition to costs associated with travel and coordination. Nevertheless, our experience has been rewarding and enriching and, as a result, we have chosen to continue our successful collaboration as a joint robot soccer with a new set of Nao robots.

Our interest in RoboCup is driven by our research ambitions in cognitive robotics and knowledge representation. As such, in moving to the Nao robots (for our first time), we chose not to port across our existing AIBO code-base but to start entirely from scratch and challenge ourselves with a new highly innovative architecture, new design and new development language. Our first exposure to the robots being just 60 days before competition when they arrived from France, we did not have the luxury of time to develop a sophisticated player, but instead aimed to build as many core competences, as 60 days of development would allow. We focused on vision and behaviour.

Despite the obvious constraints, we created a working player that was highly tunable and that was easily adaptable to the Technical Challenges; we came 5th is each Challenge, and 9th overall. For the RoboCup 2010 competition we decided to focus on defense. Our robots defensive strategies and mechanisms performed well. They played a successful defensive game and came close to scoring in many circumstances. By the end of the competition, as the challenges were occurring, we had dramatically improved our system and were particularly pleased with the results on the challenges, ranking in the Top 10.

Team Structure

WrightEagleUnleashed! is a collaboration between UTS and USTC. While both universities have an involvement in all aspects of the system, primary responsibility for tasks in 2010 were assigned to different universities:

- The development of the soccer player was undertaken at UTS. With one team member working full time, three members working fractional time (full time during the competition week) and the remainder of the team offering support and analysis throughout the effort.
- The development of the locomotion engine occurred at USTC. However, while the resulting system was an excellent research outcome, we chose not to use it in competition due to several technical constraints which became apparent in Singapore.

- The development of the iPad capability of the Open Challenge occurred at UTS, the development of the visual imitation capability of the Open Challenge occurred at USTC and these elements were successfully integrated at RoboCup 2010. We were delighted with our highly ambitious Open Challenge achievement.
- The passing and dribbling challenge were undertaken by UTS and USTC jointly. We gained fifth placing in both challenges and were only one second away from a Third Placing in the dribbling challenge.

Research Interests

Our research objectives at RoboCup 2010 were to:

1. Use the competition as an opportunity to familiarize ourselves with the Nao robots and engage with other international research teams in a common research platform.
2. To ‘experiment’ and ‘play’ in our development with the aim of producing an innovative approach to Robot Soccer. Given the choice between elegance and performance, we had a preference for elegance even though we were aware that this would harm our soccer performance in our first year, but preferred to take a strategic view and see our first year in the competition as one that put us in a good position for our second year.
3. To explore human inspired attention-based architectures for robot control in soccer matches.

RoboCup Player

System Architecture

Our robots used a single-threaded control strategy with a top-level sense-think-act control loop synchronized to the camera’s frame-rate. With each iteration of the control loop, the system performs the following steps:

1. Sense: grab a frame from the camera, collect the current joint readings and check other sensors (ultrasound, buttons, etc).
2. Think: execute the attention-driven subsystem (described in the following section) and collect attention votes for ‘behaviors’.
3. Act: perform those actions that received the highest attention as measure by the votes in the thinking cycle.

The system architecture is therefore based around three major packages of modules:

1. Sensing: modules for vision and for reading external body state into the internal system state
2. Attention-driven ‘mind’: modules for the system’s intelligent behaviors with control/choreography directed by an attention allocation scheme
3. Acting: modules for translating intended actions into physical effects

In our final system, sensing and acting were combined into a unified ‘body’ package. Note that this high-level architecture is generic. It does not include any soccer-specific processes. Soccer playing capabilities are added to the architecture by implementing modules for the ‘mind’. For example, localization is implemented as an undistinguished module in the attention-driven mind.

Body

Locomotion

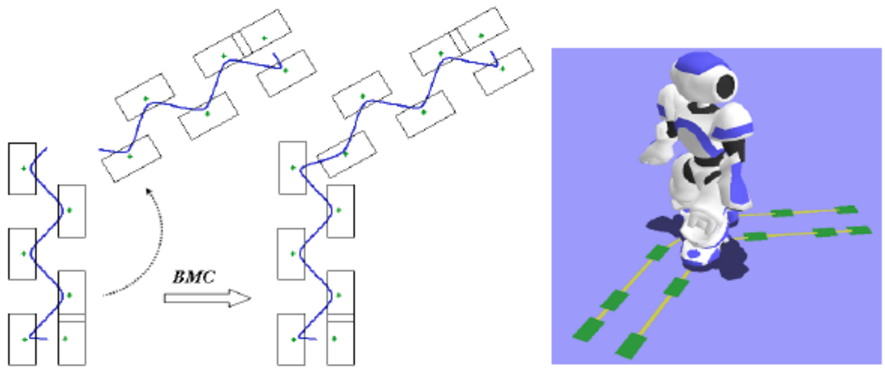
Our walk engine was highly sophisticated and largely designed at USTC by Jinsu Liu (2010). Flexibility is a key capability in playing robot soccer. Changing the motion of a biped robot smoothly and quickly in real-time is an important soccer skill and a major challenge. In a paper by Liu,J. et al. (2009), we introduced a new mechanism, namely *Simplified Walking*, to generate walking patterns. The simplified walking is specified as follows:

1) The robot is considered as a Linear Inverted Pendulum (LIP) which is supported by one of its feet. The mass of the robot is distributed on the Center of Mass (CoM) which moves in a horizontal plane with a constant height.
2) The walking process only consists of a series of single-support phases. The robot uses each of its legs as the supporting leg in turn. One simplified step is assumed to switch to the next one immediately.
3) While in a simplified step, we only use a single Zero-Moment Point (ZMP) instead of a ZMP trajectory in real biped walking.

In our approach, we first simplify the walking process. Then, we calculate walking solutions which leads the robot to the destination and also satisfies the walking constraints. For example, the ZMP should be inside the convex hull of the contact points between the feet and the floor. At last, we convert a simplified walking to real walking patterns by considering double-support phases, height alteration and multi-body physical properties. With this mechanism, the continuous motion control is converted to the decision of a ZMP in the horizontal plane for each walking step so that complex biped motions can be planned with significantly reduced computation cost.

Periodic motions are most often used in applications of humanoid robots. We define *periodic biped motions* as the biped walking motions which consist of three types of steps: *starting steps*, *walking steps* and *stopping steps*. Each walking sequence starts from a pair of starting steps and ends with a pair of stopping steps. A continuous walk can be generated by repeating a pair of walking steps.

To make the robot gain more flexibility,we use a technique, called *Biped Motion Connection*, to connect the periodic biped motions together. The connection problem is defined as inserting two simplified steps in order to achieve the connection between the current walking and the connected walking. Figure 1(a) shows a smooth connection of two CoM trajectories. The simulated robot achieved the connecting steps successfully as shown in Figure 1(b).



(a) CoM trajectory connection (b) Simulated walking

Figure 1: Biped motion connection

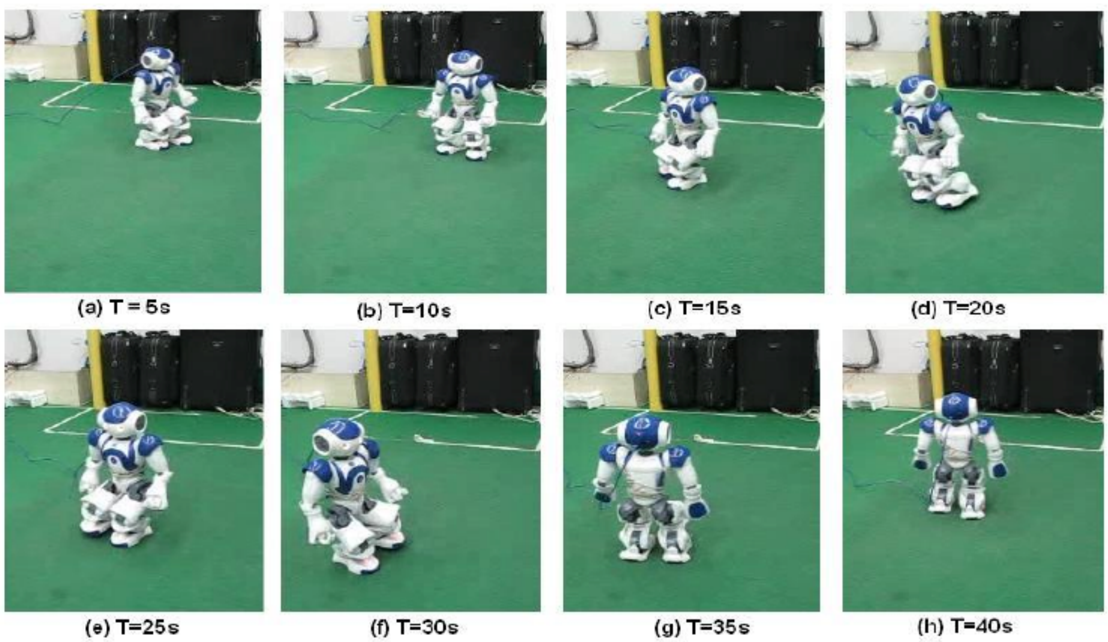
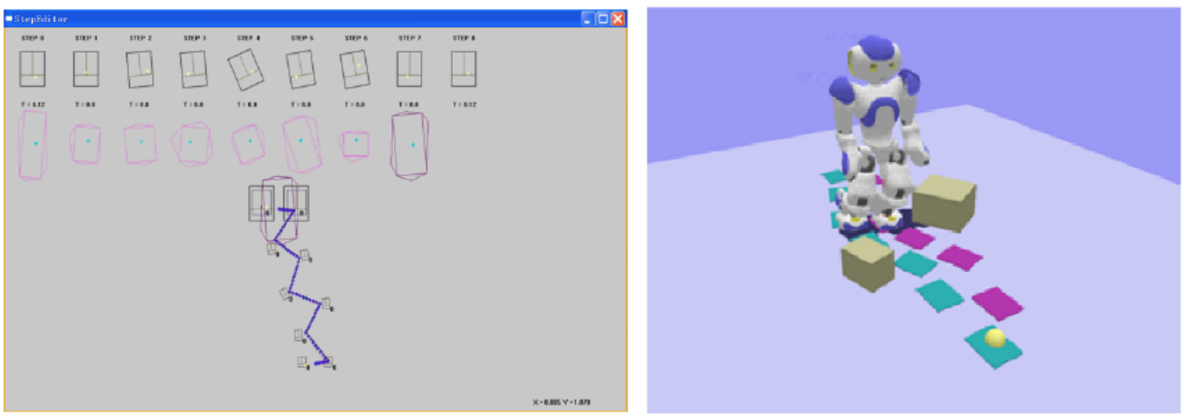


Figure 2: Online biped motion planning controlled by a joystick

With the techniques mentioned in this section, we designed several periodic biped motions including forward walking, sideways walking, turning and so on. We realized an online motion planner to generate walking patterns for a real Nao robot, while the subsequent motion can be connected to the current walking motion using biped motion connection automatically. In the snapshots in Figure 2, we demonstrate a combined motion which is controlled by a joystick.

Unfortunately, there was insufficient time to integrate and optimize the walk engine for competitive soccer, and it was only used in the open challenge. As such, throughout the competition our robots used the default walk provided by Aldebaran using the velocity-control API. Kicking behaviors were designed by hand using the Aldebaran 'Choreographe' software package and exported as Python code for integration into our player.

There have been several tools developed by USTC for humanoid motion planning. We designed a software system to edit footstep plans and generate walking patterns for Nao robots. The editor is shown in Figure 3(a). The footfalls can be designed by a mouse driver. The CoM trajectory is calculated according to the footstep plan, automatically, by using ZMP sampling search algorithm, which we first proposed in 2008 (Liu, J. and Veloso, M. 2008). All walking motions are tested on our simulator firstly, in order to make sure that the motions are feasible and do not create potential harm to the robot. The simulator shown in Figure 3(b) employs ODE(Open Dynamics Engine) for the physics calculations. The model of the robot is described by using VRML(Virtual Reality Modeling Language).



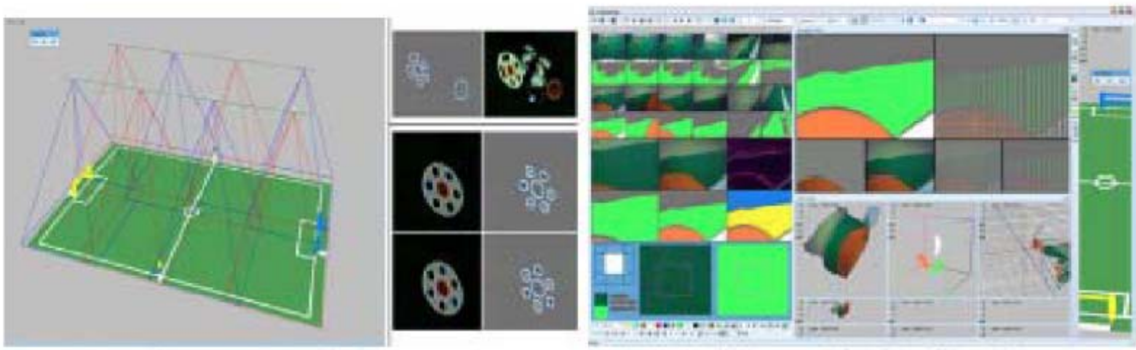
(a) Footstep Editor

(b) Nao Simulator

Figure 3: GTMS and Robot Studio for robot development

Vision

USTC and UTS jointly have an extensive range of vision algorithms and tools for analyzing vision data such as an algorithmic test platform for machine vision and a GTMS (Ground Truth Measurement System) that can be used for various kinds of important experiments. The former focuses on problems in image segmentation, clustering analysis, object recognition, three-dimension reconstruction and color learning. The latter is a low cost GTMS that uses a common USB video device, which is inexpensive but it can be used to construct the whole system (see left hand side of Figure 4(a), below). We experimented with an algorithm (see right hand side of Figure 4(a)) that was not only simple but exhibited good performance in tracking objects of interested. Only two or three personal computers are needed to run eight cameras, at a rate of nearly 30 fps each, simultaneously. No color table is required for the recognition process and the camera calibration process is automatic. Both are implemented in our project Robot Studio (see Figure 4(b)). This system is extensible and we plan to add new features as we require them. With the help of this advanced tool we are able to test a large range of possibilities practically and efficiently. Moreover, this tool supports a wide range of opportunities for undergraduates to join and contribute to our project, research and robot soccer team.



(a) GTMS (b) Robot Studio

Figure 4: GTMS and Robot Studio for robot development

After experimenting with a number of algorithms and prototypes, we chose to build the vision system upon the OpenCV open-source vision library (Bradski 2000). An advantage, in our case, of using OpenCV is that it permits rapid prototyping and a simplified code-base while exploiting highly optimized algorithms.

The vision subsystem operates according to the following pipeline:

1. An image is retrieved from the camera in 640x480 YUV format with auto-white-balance and other automatic configuration *disabled*.
2. Vertical lines in the image are then sampled (at 10 pixel spacing) to find the top-most run of field-colored pixels. Field pixels are identified by direct look-up in a preloaded 16MB color look-up table.
3. The lower convex hull of the runs of field-colored pixels is then assumed to be the field and all vision code is restricted to this part of the camera image (eliminating spurious background noise).
4. Each pixel inside the convex hull is converted to a color index (e.g., FieldGreen, White, BlueGoal) by direct look-up into the same preloaded 16MB look-up table of Step 2.
5. Isolated pixels are ignored: any pixels without neighboring pixels of the same color are set to be black.
6. For each color index, the centre and axial orientation of those colored pixels is computed by the cvMinAreaRect2 in OpenCV. This function computes the circumscribing rectangle of minimal area. The centre of the computed rectangle is used for localizing the object in space and the axes are used in calculation of relative angle and size of the object.
7. Along with the geometric properties of the color distribution computing in Step 6, additional statistics are tracked by the vision subsystem. These statistics record how many consecutive frames have contained an object (i.e., a color) or how many frames it has been since the object was last seen. These statistics allow spurious appearances or disappearances of objects to be filtered.

Attention-driven ‘Mind’

Our previous RoboCup systems have been based on state-machines and XABSL. In 2010, our code-base was centered around a new attention-based architecture. In this architecture, there is neither central coordination nor explicit inter-module control flow. The system is designed to behave as a ‘Society of Mind’ (Minsky 1988).

The system is composed of a set of modules. Each module is designed to perform a single task. For example:

1. Scan the head in a search pattern
2. Look at the ball
3. Walk to the ball
4. Look at the goal
5. Line up to kick
6. Kick
7. Update LEDs
8. Dodge a robot
9. Detect a fall
10. Recover from a fall
11. Return to ‘home’ position

Conceptually, each of these modules executes concurrently (in practice, concurrency is achieved by cooperative multitasking in which each module gets a brief time-slice after each frame collected from the camera). Each process also simultaneously attempts to achieve its goals, even if doing so is in contradiction with other processes. That is, in each cycle the robot will be, simultaneously, attempting to chase the ball, return to its home position, look at the goal and recovering from a fall.

Of course, we do not want a robot to actually attempt to perform these contradictory actions simultaneously and this is the role of attention. Each process is associated with a level of attention (stored as an integer value). Each process performs its computation and sends to the motor control subsystem a ‘vote’ for an action to be performed. The module with the current highest attention wins the ‘election’ and its action is performed on the robot body. For example, consider the situation when the robot is giving attention to offensive strategies. In this case, if the ‘walk to ball’ module is issuing a command to walk forward while the ‘return to home position’ module is issuing a command to walk backward, then the robot body will perform the walk forward action because the ‘walk to ball’ module will have higher attention according to the offensive strategy.

Attention is set and modified by both modules reflexively and also by higher-level meta-modules that modify attention of other modules. Meta-modules, such as those for setting up initial positions or for entering offensive or defensive plays, modify attention levels so that desired behaviors are more likely to gain control of the body. Lower-level modules may also modify attention levels. For example, a ‘look at ball’ module will reduce its own attention after fixating on a ball for a long time. By reducing its own attention, the robot will continue to look at the ball if that action has a great deal of attention but in ordinary play it will allow other modules to perform actions and avoid the ‘starvation’ of other modules.

The advantage of this attention-based architecture is that it eliminates the need for state-based strategic design. Instead, we can calibrate the attention levels through a ‘coaching’ process and allow sophisticated behaviors to automatically emerge. By avoiding state machines, our robots are always active and never became stuck in an undesired or passive ‘state’.

A disadvantage of the architecture is that a significant conceptual leap is required to induct developers into the attention-based mindset. This conceptual leap is somewhat analogous to the leap from imperative to declarative programming. Instead of defining the particular steps that the robot must follow, the system is designed by analyzing a problem in terms of its underlying independent behaviors.

An obvious concern is oscillations might occur when two modules with similar levels of attention are in ‘competition’. However, we did not experience this in practice and now believe that in highly dynamic environment of competition such situations are unlikely to occur for more than a brief moment.

Further information on our attention-driven architecture may be found in the work of Novianto, Johnston and Williams (2010).

Behavior and Strategic Decision Making

As this was our first year of competition and we received our robots only 60 days before the competition, we had insufficient time to implement our planned strategic decision making capabilities into our robots. However, while the system had limited strategic capabilities explicitly programmed, we attempted to ensure that the emergent behaviors of the attention-based architecture were implicitly strategic.

Localization

Given time constraints, we felt that our best chance to create a robust player was to experiment with nontraditional approaches to localization and to create a reactive system based on the emergent properties of subjective ‘localization’. We designed behaviors such that decision making would lead to the emergence of location-sensitive behaviours, even though they did not maintain an absolute world model.

Localization therefore worked according to the following principles:

1. For each object returned by the visual subsystem, the robot would use the current camera transformation matrix and the known altitude of the object (e.g., the ball and goals lie on the ground, while the waist bands are slightly above the ground) to compute the position of the object relative to the robot’s torso.
2. A robot would, by default, chase the ball simply by walking in the direction of the ball that was computed relative to the torso.

3. While chasing the ball, the robot would simultaneously attempt to line up for a kick by guiding the walk based on the angle between the goal and the ball. That is, at any given point in time, the goal is assumed to be an infinite distance away and the robot approaches a kicking position according to a known displacement from the ball from the axis represented by the direction of the ball to the goal.

We had designed reliable and straight kicks, meaning that this naive localization strategy was surprisingly effective. Of course, this strategy has many weaknesses, e.g. low reliability if the robot is inside the goal area and the ball is close to the goal line. However, given the short time-frame for development we attained reasonable performance. Given time constraints, we chose to accept the disadvantages of subjective localisation and attempt to solve these problems in 2011.

Coordinated Control

We developed a number of network-based coordination strategies. Unfortunately, we were not prepared for the technical complications associated with reconfiguring the Naos to connect to the RoboCup routers. Because we were not confident in performing this configuration under the stressful competition environment, we decided to disable the network-based activities and revert to the manual button interface. As such, no network-based coordinated control code was operating on our robots during the competition.

Soccer Challenges

Our performance in the soccer challenges substantially exceeded our performance in the soccer matches. Our significant improvement stemmed from the improvements to our soccer player that we were able to make in response to the matches and from the fact that we had more time to integrate our disparate code-bases and calibrate vision properly. This ability to rapidly adapt was the reason and the objective for using the attention-based architecture.

The attention-based architecture made it easy and straightforward to adapt to the soccer challenges. The passing challenge and dribbling challenges were implemented simply by removing competition-specific modules (e.g., kick), adding a new module for the behavior involved in the challenge (e.g., ultrasound-based avoidance) and fine-tuning the behavior values to give appropriate attention to the task at hand.

Of course, one could question the advantages of an attention-based architecture by arguing that, in a state-based player, adapting to challenges is also simply a matter of “deleting unwanted states” and “adding new states”. However, in our attention-based architecture each module is designed as a fully independent process that operates autonomously and continuously. Modules have few dependencies so that they can be dropped in and removed at will without impacting the overall reliability of the player. A newly created ultrasound-based avoidance module can be ‘dropped’ into a soccer player and, with no further configuration, immediately enable the soccer player to avoid colliding with other robots.

Open Challenge: Attention Based Coaching (ABC)

Interacting, communicating and collaborating with robots during their development and deployment is a major challenge and as a result there is enormous scope for new ideas, techniques and technologies that can be used to teach robots new skills, and to improve their performance.

Our long term vision is to transform robot soccer (and other intelligent systems) development from coding at a computer to one-on-one interaction in the field. ‘Coaching’ should not be a process of fine-tuning state machines but should be more like human soccer coaching. We should be able to direct our soccer robot to “pay more attention to the goals right now” or “hold off on chasing the ball” and we should be able to physically demonstrate to the robot how to kick, how to defend and how to respond to difficult situations.

The attention-based architecture goes some way towards making this possible. When a robotic soccer player is implemented as a collection of low-level skills orchestrated by the allocation and direction of attention, a ‘coach’ can watch a player and immediately configure the flow and allocation of attention.

Our entry in the open challenge was based around this vision. We built a real-time human-robot interaction software system based on the Apple iPad tablet computer. It enables the user to act as a trainer and, thereby, simultaneously monitor and instruct multiple robot soccer players within the actual soccer field environment. Combined with a vision based coaching system (developed by USTC), we can train the robots to perform actions based on direct physical demonstration from a person. In the open challenge, we applied this human-demonstration activity in the context of goal-keeping.

Human-Robot Interactivity (iPad Application)

Our iPad console runs a native application developed using the Apple iOS SDK. It talks directly to a communication proxy module running on the robots. The main interface of the application consists of a camera view and a soccer field view which displays raw or processed live images streamed from the robot and the position on the field that the robot believes he is located respectively (Figure 5). Properly configured robots automatically establish a connection with the iPad console once they are turned on. An arrowed dot icon is displayed in the soccer field view on the console for every connected robot. The positions of the dots in the soccer field reflects the position the robot believes it is located on the actual field. The user of the iPad console - the coach - can interrogate each connected robot by tap-selecting the corresponding dot in the soccer field and the specific operational details of the robot will be displayed in the camera view on the console. Double tapping the camera-view triggers streaming of the live images captured from the Nao’s camera. The user of the iPad console can monitor what the robot “sees” in real-time. The combination of all this information gives a human robot coach a clear picture of the mental state of the robot in real-time.

Figure 5. The iPad user interface.

The iPad console application relies the finger-tracing on the touch surface and accelerometers of the device to provide rich instructions to the robot. For example, when the robot is unable to correctly identify a known object due to object occlusion, as in Figure 5, the user can draw a circle around the blue goal using finger-trace to send an instruction to the robot to run its object identification algorithm and try to recognise the object. The robot may give back a list of possible objects it thinks the target may be. We can provide the final confirmation of the object identification. The robot can then save all associated parameters of the objects for later use. In the following case of direct action learning, by tracing around an human demonstrator, we provide additional assistance for robot to identify the human from the background environment and hence greatly improve the efficiency of human-action learning.

Direct Goal-keeper Coaching (Visual Imitation)

The visual imitation demonstration is an early show-case and application of an ongoing research project underway at the University of Science and Technology of China. Details of the project will be forthcoming in future publications or by contacting the Multi-agent Systems Laboratory at the University of Science and Technology of China. The demonstration was built upon parts of OpenCV and basic operation of the system can be understood as operating according to the following pipeline:

1. A model of the background is learnt on system start-up (and this model is maintained throughout the demonstration)
2. For each frame, background substitution is performed to extract the shape of the ‘coach’
3. A geometric model of the human body is matched against the observed outline of the body
4. The geometric model is translated into Nao joint commands
5. The Nao joint commands are transmitted to the robot for imitation

All computation in this demonstration is performed on-board the Nao and using the Nao’s camera at a resolution of 320x240. That is, there is no off-line processing.

Conclusion

We were a new team in 2010, our robots arrived 60 days before the competition and during that time we developed an entirely new and highly innovative system inspired by the role of attention in decision making during human soccer matches. Our robots performed well in the soccer competition and achieved a fantastic result in the Technical Challenges gaining 5th place in each one giving rise to a 9th placing overall.

As a result, RoboCup 2010 was, for us, an opportunity to test our ideas of attention-driven behavior control in a complex, real-world setting. While we did not have the time and resources to develop solid low-level skills, we were able to create a system that enabled us to rapidly react to the conditions of competition and create a platform for future research and a more competitive team in 2011.

References

Bradski, G. (2000) ‘The OpenCV Library’, *Dr. Dobb’s Journal of Software Tools*, pp. 120-126.

Karol A. and Williams, M-A. (2006). Distributed sensor fusion for object tracking. In *Lecture Notes in Computer Science, Volume 4020, RoboCup 2005: Robot Soccer World Cup IX*.

Liu, J. (2010) *Biped Motion Planning for Humanoid Robots in Complex Environment*, Chinese-language PhD Thesis, University of Science and Technology of China.

Liu, J., Chen, X., and Veloso, M. (2009). Simplified walking: A new way to generate flexible biped patterns. In *Climbing and Walking Robots and the Support technologies for Mobile Machines, CLAWAR'2009*, Istanbul, Turkey.

Liu, J. and Veloso, M. (2008). Online ZMP sampling search for biped walking planning. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'2008*, Nice, France.

Mendoza, R., Johnston, B., Yang, F., Huang, Z., Chen, X., and Williams, M-A. (2007). OBOC: Ontology based object categorisation for robots. In *The Fourth International Conference on Computational Intelligence, Robotics and Autonomous Systems 2007*.

Minsky, Marvin *The Society of Mind.* Simon and Schuster, New York. March 15, 1988.

Novianto, R., Johnston, B. and Williams, M-A. (2010), 'Attention in ASMO cognitive architecture', *First International Conference on Biologically Inspired Cognitive Architectures (BICA)*, Nov 12 - 14, Arlington, United States of America.

Open Dynamics Engine (2000). http://www.ode.org.

Stanton, C. and Williams, M-A. (2005). A novel and practical approach towards color constancy mobile robots. In *RoboCup 2005: Robot Soccer World Cup VIII Series: Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence*.

Virtual Reality Modeling Language (1993). http://www.vrmlsite.com.

Zhang, X., Chen, X., Li, J., and Li, X. (2006). Vision-based Monte Carlo - Kalman localization in a known dynamic environment. In *International Conference on Control, Automation, Robotics and Vision. (ICARCV'2006)*