The Northern Bites 2007 4-Legged Robot Team

Henry Work
Prof. Eric Chown
Jeremy Fishman, Johannes Strom, George Slavov
Mark McGranaghan, Tucker Hermans

Department of Computer Science
Bowdoin College
8650 College Station
Brunswick, ME, 04011-8486, USA
hwork@bowdoin.edu
echown@bowdoin.edu
`http://robocup.bowdoin.edu`

February 14, 2007

Abstract

This document serves as the team description paper for the purposes of applying to qualify for the RoboCup 2007 4-Legged League World Championship competition in Atlanta, USA.

1 Intent to Participate

Bowdoin College's 4-Legged RoboCup Team, the Northern Bites, intends to participate in the 2007 RoboCup World Championships. As a small, liberal arts, undergraduate college, Bowdoin is not the typical entry in international robotics competitions. However, the lessons learned by qualifying for and participating in RoboCup 2006 and finishing tenth place in our first year of international competition has made the Bowdoin team more formidable than ever. The success of last year's team has increased interest from students and more than compensated for the loss of five graduating team members. Full-time development for the 2007 season resumed in July of 2006 – less than a month after RoboCup 2006 – and has continued at a fast pace in the hopes of improving our code base and producing innovative research within the RoboCup community. We are applying this year as a second-year team, after a very successful first year, to test our mettle once again with some of the best research universities in the world. RoboCup 2006 was a tremendous learning experience for us, and we have spent the year since working towards competition at the German Open this April and RoboCup 2007 in July.

2 Team Information

Professor Eric Chown founded the Northern Bites in the spring of 2005. The team grew out of a fall 2004 *CSCI 320: Robotics* course. In its infancy, the team had only one student member who worked on it as a senior honors thesis [8]. That version of the team competed at the 2005 U.S. Open. What was unusual about the team was that all of the code was written by only one student. None of the code came from other teams.

The 2006 team brought with it a completely new code base and a much bigger development team. The project gained momentum throughout the summer and fall within and outside of the CS department.

Progress picked up greatly in the spring of 2006, about six months before RoboCup in Germany. The team qualified for the 2006 World Championships. They started a weblog to document and discuss their progress on the project that is still active to this day: `http://robocup.bowdoin.edu/blog/`. Seven students took independent, accredited studies working on the development for the 2006 team and were joined by eight or so more members who contributed in their spare time. The team held an on-campus demonstration in April with over 100 spectators in attendance. The team took 12 members to the US Open in April 2006 and won two games. Furious development resumed in preparation for RoboCup 2006 in Bremen, Germany. In the end, the Northern Bites ended up finishing in what amounted to a tie for 10th place, including a tie to a top 8 finisher and loses only to teams in the top 8.

RoboCup and the Aibo platform have become a central part of the Bowdoin Computer Science curriculum, meshing well with the department's strengths in cognitive science and artificial intelligence. Using software developed at Bowdoin called AiboConnect [1], the Aibos have been put to use in virtually every course in the computer science curriculum including courses such as Data Structures, Programming Languages, and most recently, a Robotics course in Fall 2006.

RoboCup serves a number of purposes at Bowdoin. 1) It is a platform to attract students into the discipline. 2) It is a way of getting students interested in research very early in their careers (this is important as Bowdoin has no graduate program). 3) It touches on a number of research areas related to work done by Eric Chown. 4) Finally, and most importantly, it is a great way for undergraduate students to learn about Computer Science. It is a wide ranging and interesting project with something for just about everyone. This is the main reason that we have not used other team's code. Undoubtedly we could do so and get better quickly, but it doesn't serve the purpose of our institution, which is to teach students how to do all of these things on their own. In the short run this may have been a limitation, but ultimately it gave our students ownership in the team and a greater sense of purpose.

3 Technical Report

Northern Bites is committed to implementing its own solutions to the problems faced in RoboCup. Last year a tremendous amount of work was put into getting the team functioning and building the basic elements of any RoboCup team: a slew of debugging tools, a dynamic behavior architecture, and of course, the motion, vision, localization, and communication systems themselves. Our goal, quite frankly, was merely to be able to score on the worst teams. Based on last year's success we have much loftier goals this year. We wish now to be competitive with the best teams and to engage in innovative research. Last year we did not participate in any of the challenges; this was mainly due to the sheer amount of development required to get the dogs to play and to be competitive in some of the games. This year we plan on participating in all three challenges.

Our code base this year has been built directly from last year's team. The improvements this year include: a full python framework for behaviors and localization that runs at 30 fps, a completely new approach to localization, a heavy focus on improving development via offline vision, localization, and behavior debugging tools, and a completely original approach to vision processing.

3.1 Vision

Our vision system has evolved tremendously over the past two years. The original system consisted of a vertical scan-line algorithm that performed recursive blobbing to identify objects. Simple hand-tweaked threshold values were found via trial-and-error to threshold the eight 4-Legged League colors. This approach was slow and inaccurate. Our vision system was scrapped early last year in favor a run-length encoding system and a new calibrating tool that uses edge detection to greatly simplify the mapping of colors. This tool produces far better separation between colors and reduces the time of calibration to under an hour (a 75% calibration takes ten minutes). Last years team also implemented rUNSWift's chromatic distortion filter to reduce distortions ('the blue ring of death') present in the Aibo's cameras.

Since June there have been a number of important developments in the vision system. We've developed an offline debugging tool, dubbed Cortex, that allows us to work on vision algorithms with full, unprocessed (but compressed) images from the dog. This helps us work on color mapping and chromatic distortion filtering as well. Cortex has been a huge boost for development. We've added an accurate line and corner detection system that is fast and reliable for use in localization. We now also estimate the pose of the robot. Through this we are able to produce an estimated horizon line for vision processing, translations for distances and angles to center of the robot's body, and have the ability to project distances and angles to corner line points.

Most importantly, we are integrating a vision system that is very cognitive in nature. Bowdoin started a Robotics program because of Professor Eric Chown's interests in Cognitive Science, specifically in Cognitive Mapping, the human vision system and their impact on spatial processing. While the performance demands of RoboCup prevent a truly cognitive approach, it is our belief that cognitive ideas can easily and usefully find their way into practical robotics [2]. The ideas that we are trying to bring into our vision system are based upon the cognitive ideas of attention and short-term memory. In a nutshell, one of the things that humans are very good at is whittling down the vast amount of information available into a few manageable chunks. Whereas robots usually relentlessly process most or all of the visual field, people generally focus on a small area and efficiently find the key places to look. Our new vision system is fairly simple in principle and in execution: the idea is to use information memory, the shape of the field, its layout, etc. to guide the search for landmarks, balls and other important objects. When it works well the vision system can find the important objects with very little processing and can actually leave a huge amount of the visual field unchecked. In addition, since we are pursuing an active approach to vision we do not use the traditional run-length encoding style algorithms used by many RoboCup teams. Instead we basically explore the visual field looking for objects and when we find good candidates we essentially perform the sorts of checks that teams do as an integral part of the scanning process.

3.2 Localization

Our localization system in 2006 was the least successful of all our endeavors. The 2006 team used a modified version of Monte Carlo Localization that performed very slowly (our initial system slowed the dog's vision updates to 4-5fps!), had no odometry estimates, and relied on poorly estimated distances and angles from visual landmarks. We were not able to self-position reliably and we used localization very little in our behaviors in Germany.

Our goal since RoboCup 2006 has been to do everything possible to establish a solid localization system that is fast and accurate. We've learned from Germany that reliable localization was a clear separator between the top and bottom tier teams. In the fall, we chose to implement an Extended Kalman Filter based on the NUBots successful implementation [3]. An EKF was chosen because it is fast, accurate, and deals with noise very effectively, criteria which we believe are essential in creating an elegant solution to the localization problem. Something we are trying that is different has been to write the system entirely in Python. This is possible thanks to a team-written C-optimized matrix library that is efficient enough at matrix computations to run at 30fps. Writing the localization system in Python has been a joy because it has been easy to debug and change on the fly. To speed up development even further, we've written an off-line localization debugger, that runs vision and odometry log files created on the dog, allowing us to replay, debug, and tweak parameters using real, noisy data.

3.3 Communication

Last year we implemented a UDP-based system that broadcast information at approximately five packets a second. Without a reliable localization system, the only useful information communicated was distance to the ball which was used for simple role-switching. This made it so that two of our own dogs didn't clog the approach to a ball and it worked surprisingly well for such an unsophisticated method. This year we are using communication for positioning, role-switching, and shared ball estimates. We subscribe to the idea that there should be no negotiation or team-captain leadership between the dogs, as this can often lead to dogs who refuse to approach the ball and develops problems in a packet-loss environment such as RoboCup.

3.4 Motion

In the Summer of 2005, because of the time and resources necessary to write an effective walk engine, the team spent a few weeks implementing an engine developed for the UPenn 2005 team [4]. We have since built our own kicking and special move engine along with an easy debugging tool. After Germany this summer, we spent time optimizing our walk engine using simple machine learning methods (hill climber, policy gradient) and achieved only marginally faster straight walks. Our fastest hand-tweaked walk has achieved a speed of nearly 40 cm/s. Our research with motion is now directed towards finding omni-directional walks using an overhead camera [5] as well as optimizing our grab-walk [3]. In the fall, we added odometry calibration for orthogonal and spinning directions which are automatic and can be repeated at competition should field differences hamper our walk. The odometry data is used both on the behavior and localization levels.

3.5 Behaviors

A substantial improvement last year was integrating a Python layer using bindings and a port of Python 2.3.3 from the rUNSWift 2005 code base [7]. This sped up development on behaviors by an order of magnitude, but unfortunately the system wasn't ready until two weeks before the RoboCup 2006. The entire set of goalie and player behaviors used in RoboCup 2006 were written one week before competition (though many behavior concepts hammered out while developing in C++). Needless to say, there is room for tremendous improvement this year with a functional localization system and a much better platform (Python) for development. We have also written a 2-D simulator to be able to prototype simple behaviors for faster development without wearing down the Aibos.

What we learned from last year is that the fundamentals of the game are extremely important. A good team gets to the ball fast, grabs the ball with great consistency, and then moves the ball down the field in the right direction. These were the sum goals of our team last year and we were somewhat successful because of them. Powerful kicks, over zealous role switching, and heavy reliance on localization that we saw in other teams resulted in balls flying out of bounds (or worse, dog's breaking), dogs refusing to approach the ball, or grab gaits that couldn't see the opponent's goal. Our goals this year are much loftier and will increase our reliance on our low-level systems; however, we do not want to lose sight of the fundamentals: we have to be fast, be able to effectively grab the ball, and then move it down field in the right direction. Thus we are working hard to greatly increase the consistency of our grab. We want every dog to know where the ball is at all times on the field – something we failed miserably at last year. We want to figure out the right strategies to move the ball down field while avoiding obstacles. We are investing a lot of research in goalie dodging and optimizing our grab-walk so that we move fast with the ball and not just to the ball.

We also want to bring a more cognitive bent to behaviors. While it is true that low-level behaviors currently differentiate the best from the rest in RoboCup, what is really exciting is to see team play. Last year we saw that a team that was essentially purely reactive could achieve a fair amount of success in competition. It stands to reason that a team that can incorporate more high-level intelligence and team play can do even better. Since our low-level systems are finally rounding into shape we now have the ability to work on some of these ideas. Among the ideas that we hope to bring to Atlanta are: 1) Set plays on kickoffs. Kickoffs are the most obvious place to start when considering developing plays since the positions are fairly static at the beginning of the play. 2) In game learning. We plan to track our performance during the game on a variety of measures. Based on those statistics we should be able to alter our strategy on the fly without the need for timeouts or stick switching. For example, our team might notice that an aggressive strategy is not working and automatically switch to a more defense-oriented approach. 3) Coordinated behaviors. Players should be able to work together better than we have seen from most teams. For example, many teams shoot the ball virtually whenever they get it. Such long-range shots should never score. A coordinated goalie and defender should be able to easily screen the entire goal. Conversely, on offense one player might try and intentionally screen the goalie's vision. Such strategies are basic to sports like hockey but we have seen scant evidence of them in RoboCup. Part and parcel of all of this is that players have a larger sense of the game than what their sensors tell them. Again, this is an area that plays directly into the strengths and research of the team's adviser Eric Chown.

4 Statement on Code From Other Teams

The decision was made early in the summer of 2005 that we would provide an original and fresh code base and would not directly rely on other teams work to propel our team forward. This decision was grounded in two main ideas. One, the RoboCup development cycle provides an incredible learning experience (and if original, is way more fun) for our undergraduate team members. Two, we believe that providing our own unique solutions and innovative research to the robotic community is in step with RoboCup ideals. That being said, there are three significant code pieces used from other teams.

- Inverse kinematics walking engine from University of Pennsylvanias 2005 code base [4]. The team also looked at examples from UNSW 2004 code base [6] to interact the joints with the OPEN-R SDK.
- A Python 2.3.3 port along with C++/Python bindings created by the rUNSWift 2005 team [7]. We also use Makefiles and memory-stick shell scripts inspired by the rUNSWift 2005 team [7].
- Chromatic Distortion Filter written by rUNSWift 2005 team [3]. We use their calibration tool but have built a GUI on top of it to enable further research.

5 Relevant Publications

1. Chown, E., Foil, F., Work, H., Zhuang, Y. (2006) AiboConnect: A simple programming environment for robotics. To appear in the Proceedings of the 19th International FLAIRS Conference.
2. Chown, E. and Boots, B. (2007). Learning Cognitive Maps: Finding Useful Structure in an Uncertain World. To appear in Jefferies, M., Yeap, A. (eds.) Robot and Cognitive Approaches to Spatial Mapping.
3. Quinlan et al. (2005). The 2005 NUBots Team Report. University of Newcastle.
4. Buchman, G., Cohen, D., Vernaza, P., Lee, D. (2005). The University of Pennsylvania RoboCup 2005 Legged Soccer Team Report.
5. Hebbel, M., Nistico, W., Fisseler, D. (2006). Learning in a High Dimensional Space: Fast Omnidirectional Quadrupedal Locomotion. RoboCup 2006 Symposium.
6. rUNSWift 2004 code release.
7. rUNSWift 2005 code release.
8. Foil, G. (2005) Implementing a robocup team at a small liberal arts college. Honors thesis, Bowdoin College.

6 Video Submission

Five months away and the Northern Bites can't wait for RoboCup 2007. Our video is posted here:

`http://robocup.bowdoin.edu/media/`

7 Code Diff

Because our team uses limited code from other teams and is not based on any team's framework or architecture, we do not believe a diff is necessary.