Cognitive Visuo-Spatial Reasoning for Robotic Soccer Agents

An Honors Project for the Department of Computer Science

By Elizabeth Catherine Mamantov

Bowdoin College, 2013

Contents

List of Figures

Abstract

This project explores the application of theories of human navigation and spatial cognition to the problem of robotic localization, which is the way that robots determine their location within an environment. It does so in the context of RoboCup Standard Platform League competition, in which humanoid robots are programmed to autonomously play soccer, typically using a probabilistic filter to localize themselves on the field. An alternative method of localization based on cognitive principles is presented and contrasted with more traditional techniques, and three contributions toward the goal of creating a cognitive localization system are described: a non-probabilistic goalie agent, a system to project visual objects into simulated images, and a visual memory system. The strengths and weaknesses of these developments are explored in the context of the debate over the applicability of cognitive theory to robotics.

1 Introduction

Robotics has become an increasingly important proving ground for artificial intelligence (AI) because an AI program controlling a mobile robot is forced to confront the real world and attempt to act productively despite the overwhelming amount of often contradictory data. While early AI researchers solved problems using high-level symbols and assumed that "easy" tasks such as object detection or speech recognition would be solved later, robotics has shown that the problems that seem the simplest to humans are often the hardest to solve with software and hardware.

Playing soccer is a wonderful example of a task that is cognitively very simple for human beings; compared to chess or logic puzzles, most people would say that soccer is an easy game. However, while computers can defeat the greatest human chess players, robotic soccer is a dauntingly difficult proposition. A human soccer player does not have to think twice about what it means to follow the ball or position as a defender; built-in cognitive capabilities that have evolved over millions of years provide people with amazing tools to enable them to act in the real world.

As robots have none of these abilities without sophisticated programming, robotic soccer is an extremely productive research testbed when it comes to many different areas of AI; computer vision, navigation, strategy, team cooperation, and motion are all pieces of a functioning robotic soccer agent. RoboCup, an international robotics competition, has many leagues that allow researchers to attempt to solve the problem of robotic soccer in different ways, as well as other competitions that focus on other types of robotics. The main goal of RoboCup is to have a robot team play the human world champions by the middle of this century [10], and thus it fosters research into the various areas that relate to soccer-playing agents.

Bowdoin's RoboCup team, the Northern Bites, competes in the Standard Platform League (SPL), in which all participating teams use standard hardware. While some leagues can solve problems by equipping their robots with sophisticated sensors or other specialized hardware, in the SPL, all teams use Aldebaran's Nao humanoid robot. This robot provides basic sensors—two video cameras and sonars for detecting the environment, plus an inertial unit, force sensitive resistors, and joint sensors for sensing its own body—and teams are not allowed to modify their robots in any way. Thus, the focus of the SPL is on the software controlling the robot, and the most successful teams are those that utilize the shared sensors and enactors in the most intelligent way via their control programs.

The focus of this project is the problem of self-localization, which plays a large role in the effectiveness of in any robotic soccer team. For a robot to play soccer well, it needs to know where it is located on the field; this knowledge plays a large role in the robot's behavioral decisions. For example, when close to its own goal, a robot needs to act defensively and kick the ball away from the goal, but if it is close to the opposite team's goal, it needs to shoot. Based on the human experience, this should not be a difficult problem to solve; few if any human soccer players become confused enough to shoot on their own goal on purpose. However, localization is one of the seemingly simple problems that robotics has shown to be extremely difficult in reality. Humans developed their navigation-related processing over an evolutionary time scale, and robotics algorithms have struggled to replicate the accuracy of human spatial cognition.

The accepted methods of robotic localization generally utilize probabilistic filters that take into account noisy data and imperfect sensors in order to estimate a robot's position in an environment. While these algorithms are widely used and often present acceptable solutions to the localization problem, it is worthwhile to note that they have no basis in theories of human spatial cognition, despite the fact that humans consistently outperform even robots equipped with high-tech sensors that provide very detailed information about the environment. This discrepancy suggests that applying knowledge of human cognition to robotic localization and navigation would be helpful.

Nonetheless, humans and robots process information in very different ways, so simply transposing theories from cognitive science into robotics does not work. The ways in which cognitive theory can inform robotics, particularly in the context of RoboCup, are still up for debate; this project attempts to address some of the questions related to the application of cognitive theories of localization to soccer-playing robots. Of course, the cross-pollination between the fields of robotics and coginitive science goes both ways; this implementation of ideas from human cognition requires many details to be explored and filled out, whereas a purely theoretical understanding could get away with being more vague. Thus, the long-term goal of this research is to allow robotic visuo-spatial processing to approach a more human level, which would be beneficial to soccer-playing robots and any other agent that must interact with the real world in a meaningful manner, while also shedding light on the cognitive theory that inspires the robotics side.

As a step toward this goal, a cognitively-inspired soccer goalie was created that does not utilize a traditional probabilistic localization system but rather bases all of its behaviors on direct visual perception. While this demonstrates that alternative localization techniques can be viable, the goalie also has weaknesses that further the discussion of the suitability of cognitive techniques to robotic soccer. Furthermore, some techniques that could be used to expand the capabilities of non-probabilistic localization—a visual projection system and visual memory—were also developed as part of this project and play a significant role in its goal of ultimately creating a cognitive localization system for soccer agents.

2 Background

The problem of localizing a mobile robot in a known environment is well-studied and has several standard solutions. In this section, two of these techniques are presented: the Extended Kalman Filter (EKF), which has been used in the Northern Bites platform for self-localization in the past [13], and the particle filter, which is currently in use for competitions. While a detailed mathematical description of both of these algorithms is beyond the scope of this paper, a high-level overview is helpful to the following discussion of alternative approaches to localization. These descriptions draw heavily on [16], a standard textbook in the field. These probabilistic techniques will be contrasted with a model of the cognitive system that underlies human localization and navigation.

2.1 Extended Kalman Filter

The EKF is a common solution to the localization problem and has been used in many situations in robotics, including RoboCup [14], [6]. It is a Gaussian filter, meaning that the robot's position estimate is represented by a normal distribution parameterized by its mean and covariance. Thus, there is a single "best estimate" of the robot's position, with the normal distribution around it accounting for error. In a basic Kalman filter, it is assumed that state transitions are linear, but the EKF is more commonly used because it relaxes this restriction and can approximate nonlinear transitions by using a Taylor expansion.

There are two main steps to the EKF algorithm, prediction and correction. Starting with the position estimate for time $t-1$, a prediction is made for time t based on a control vector, which in RoboCup terms is the odometry estimate given by the motion system. That is, during the iteration of the algorithm at time t, the motion system inputs an estimate of how far the robot has moved since time $t-1$, so the Gaussian location estimate can be shifted based on the vector that motion provides. Since odometry is not perfect, however, this step increases the noise in the position estimate, spreading out its distribution to account for motion sensor inaccuracies.

The correction step brings the uncertainty back down by incorporating "measurements" of the environment into the estimate. For robots on the soccer field, a measurement is an observation of a recognizable field landmark, such as a corner or goalpost; this visual input essentially serves as a corrector for motion error. This means that as the robot moves around the field, uncertainty in its position estimate will grow as the EKF simply tracks motion but will decrease when a useful field object is sighted. In this way, the EKF can follow the robot's motion around the field and correct for noise in motion and unexpected interference using visual observations.

One weakness of the EKF is that, while it is capable of tracking the robot's displacement from a known starting location, it cannot recover if the robot is placed at an unexpected location. This issue is known as the "kidnapped robot problem," and it may not seem like a major downside in RoboCup given that the robots start the game in a known position. However, robots often experience similar problems because of the unexpected nature of the game; for example, a referee may pick up a robot and move it unintentionally. Falling down also causes major problems for localization since it is so unpredictable. A fall can cause a robot to move significant distances in ways that odometry cannot track, especially in terms of the robot's heading, since a fallen robot may stand up facing in any random direction depending on how it fell down. For these reasons, the EKF is not robust enough to handle all aspects of soccer playing, and while several teams use more sophisticated permutations of the Kalman Filter in their localization systems [4], the EKF is seldom used as a standalone solution to RoboCup localization.

2.2 Particle Filter

A more robust algorithm which is currently used as the Northern Bites' main localization system is the particle filter, also known as Monte Carlo Localization. This algorithm represents the robot's location estimate as a set of particles rather than a Gaussian distribution. This offers a clear advantage in that this representation is inherently multimodal; that is, if the robot's observations suggest that it could be at one of several places on the field, the particles will be distributed around those places. This possibility cannot be expressed by the EKF's Gaussian estimate.

The main idea of the particle filter is that every particle is a "hypothesis as to what the true world state may be" [16]. Every particle has a weight, where more heavily weighted particles are more likely to be correct hypotheses. The localization system running at time t takes the set of particles from time $t-1$ and constructs a new set representing the robot's location estimate once motion and visual observations from time t have been taken into account.

The algorithm creates the updated set of particles in three steps, given the input of a control vector—in RoboCup, the odometry estimate—and the visual measurement. First, for every particle from time $t-1$, a new particle is made which represents a hypothesis for the updated belief given the estimated motion vector. Second, the weight of the new particle is calculated based on the visual observations; particles that represent locations where the given set of observations is likely will be weighted more heavily. Once these calculations have been computed for all of the particles, a third step, known as "resampling," selects, with replacement, a new set of particles from the current set. The key to resampling is that the likelihood that a particle will be chosen is given by its weight. Correct particles will likely be duplicated, whereas incorrect particles may be eliminated altogether.

Note that because the particle filter is not constrained to representing a Gaussian distribution, it can easily be extended to cope with the kidnapped robot problem. When the current distribution becomes obviously incorrect, particles representing many different field locations can be randomly inserted into the current set. If one of these particles is close to the robot's actual location, it will gain importance and ultimately the filter may reconverge to the correct position. For example, in the event that the robot becomes penalized and is placed back in play at one of two places, particles can be injected at both of those places and the filter will theoretically recover when one of the two locations is confirmed by visual observations.

While this ability makes the particle filter more robust, it also makes it more likely that the robot's localization system will converge to an incorrect field location. If a robot falls down and its particles scatter, a few incorrect or ambiguous visual identifications can cause the filter to reconverge to a plausible but mistaken estimate—a problem from which almost all probabilistic filters suffer. The algorithm is particularly likely to converge to the mirror of the robot's location on the opposite side of the field, since, with the fully symmetric soccer field, the same set of visual observations can occur at two places on the field. This is a serious problem in the SPL, and while many teams have partial solutions (for example, see [15], [4]), it is common for a "flipped" robot to attempt to score on its own goal.

2.3 Cognitive Maps

Since most RoboCup teams have implemented an EKF, particle filter, or some other probabilistic localization system and many compete successfully, the above solutions can clearly be engineered to work relatively well in the RoboCup environment. Nonetheless, there are some issues that point out that even an extremely sophisticated filter cannot match the navigational power that a human being brings to bear when playing soccer. The most obvious of these issues is the flipping problem, where a robot's localization estimate is a mirror of its actual position; this issue has led many SPL teams to score own-goals.

In contrast, human soccer players have no trouble dealing with a perfectly symmetric soccer field, and people generally localize and navigate much better than a robot in any environment. For this reason, there have been many attempts to understand human navigation and apply insights to robotic function. One theoretical model of human wayfinding is PLAN (Prototypes, Location and Associative Networks), which was developed in [8] and extended into C-Plan for use in a robotic system based on sensing corners in a building [7]. PLAN introduces several principles of human cognition that could be applicable to a robot navigating the soccer field.

PLAN builds on the concept of the cognitive map, which was first proposed in [17] and is thought to underlie most of human thinking but is particularly well-suited for navigational tasks. In the context of space and wayfinding, a cognitive map is often thought of as a set of landmarks that form nodes, with connections between them based on their spatial relationships. Because humans have limited storage and computational capabilities, these

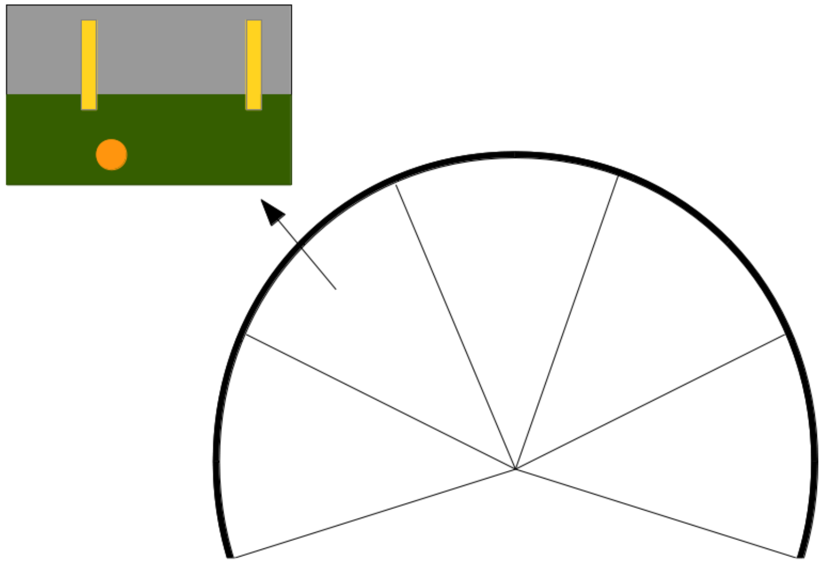

Figure 1: A representation of a local map, where the “pie” represents a human or robot standing in one place and moving its head to different directions, thus producing (in this diagram) five unique scenes.

maps are relatively sparse descriptions of the physical map and therefore scale well to large and complex environments. By carefully filling a cognitive map with the most relevant and useful information provided by the environment, humans form a solid basis for navigation without the metric detail that many robotic mapping and navigation algorithms rely on.

While physical landmarks may seem like the obvious choice for the nodes of a cognitive map, PLAN suggests that this type of map is only part of the story. It introduces a different structure called a local map, which forms the nodes in a complimentary type of map that plays a significant role in navigation and provides more specific directional information to supplement the topological structure in the landmark map. A local map is a view of the environment created by standing at a particular location and turning one's head in all directions. This creates a pie-like structure, where each “slice” is one head position; this is diagrammed in Figure 1. Each head position corresponds to a particular visual scene, which can be stored as a two-dimensional summary of the objects present in the visual field.

Note that if local maps were created for every location that a human or robot passes while navigating an environment, even systems with large amounts of memory would soon be overwhelmed and the maps would become unusably dense. Thus, along with ensuring that the stored information is abstracted from the sensory input that produced it and as compact as possible, PLAN proposes that local maps are only created and stored at particularly important locations: “gateways.” Gateways are places in the environment where a view suddenly opens onto a new region and the agent has the option to either stay in its original area or move to the new one, so gateways are important from a decision standpoint and are also visually distinctive. If a gateway is visited repeatedly and therefore useful for navigation, a local map will be created there. This network of local maps made at only the most useful positions in an environment can form an extremely space-efficient but also navigable description of the environment.

Many of these ideas are surprisingly applicable to the soccer field. In particular, the abstracted and stored viewpoints of local maps represent an intriguing way to encode knowledge of the field's appearance. Since the Nao robots are humanoid, their local maps would look extremely similar to Figure 1 because they can stand in place and pan their heads to different locations much like a human would. While the gateway concept is less obviously relatable to the soccer environment because it is completely open with no visual obstructions, there are certain viewpoints that represent behavioral gateways that are particularly important to the robot's strategy and would be clear choices for storing local maps. For example, the penalty position (see Figure 2) represents a place where the robot is suddenly put back in play after not being allowed on the field. With a judicious selection of behavioral gateways, a useful map of the field can be created.

Of course, there are many related issues that need to be addressed before such a system could be useful for the RoboCup robots, mostly related to the actual implementation of the ideas from PLAN. For instance, the details of how the gateway scenes could be created, stored, and used are not specified in their theoretical description. Similarly, the question of how to obtain concrete directional information to provide to the motion system based on a comparison of actual and stored scenes is not easily answerable. Because of the ambiguity and experimental nature of such a system, this project attempts to address some of these questions.

2.4 Comparison of Probabilistic and Cognitive Localization

Now that the basic ideas behind both probabilistic localization filters and spatial cognitive maps have been presented, a comparison of the two is warranted. While probabilistic techniques clearly have strengths and have been successfully used for many robotics applications, a cognitive approach offers its own promises.

The most obvious advantage of a probabilistic filter is that they are tried and tested; there are accepted textbook implementations of both the EKF and particle filter which have predictable performance in typical situations. In a situation such as RoboCup, where the environment has been created to benefit the robots acting in it, these solutions can be programmed to work extremely well. Many SPL teams play very successfully using probabilistic localization techniques that fail only occasionally—a few times per game is typical—because they are so carefully engineered to the environment of the RoboCup field.

On the other hand, such techniques do not scale well to larger, messier environments that humans cope with on a regular basis. Despite all of the advantages the RoboCup setup provides to the robots—an environment which is known, unchanging, and very simple, with fixed colors and lighting—localization is still an extremely hard problem, and even the best SPL localization systems can fail. These failures occur seldom enough that in the RoboCup environment it may not make sense to try to fix something that generally works well, but robotics in general does increasingly deal with situations such as messy homes or crowded supermarkets. People handle such environments effortlessly, and perhaps if robots processed space more like humans do, they would also be able to function in increasingly complex spatial situations.

In terms of actual processing, it is noteworthy that probabilistic techniques take in observations one at a time rather than having any concept of a scene. Once a corner or post has been identified by vision, it is passed into localization as an independent object, although its context is actually extremely important and certainly affects human processing of the same situation. In the PLAN model, scenes play a significant role, so an agent can consider a visual grouping of objects in terms of both a complete whole and individual objects.

A probabilistic system also does not encode its observations in a way that could be used to support spatial reasoning along the lines of human navigation; the current location estimate is the only sort of history that the program holds. That is, a human moving around an environment may remember which objects were seen most recently and thus be able to compare scenes more effectively. If the only representation of the current state is a probability distribution, there is little information for vision-based predictions, for example, which humans often use when navigating familiar paths [12].

This point gets at the heart of the differences between cognitive and probabilistic localization. In a probabilistic system, since the only history available is the current state, the robot can only rely on the mathematics of its localization algorithm and assume that the state is accurate and will continue to be accurate. It does not have the luxury to incorporate reasoning—for example, if it is not moving, then its state should not be updated—but rather runs its update methods blindly; there is no room to explicitly consider that errors might have occurred and that the state has become corrupted.

In contrast, a cognitive system does not have to rely solely on a mathematical model because it stores enough information to reason about its environment. A cognitive localization system's history is actual memory—as in the human brain—which can provide sanity checks for cases where a probabilistic system would fail disastrously. In other words, a cognitive system can reason about the localization conclusions it makes and change them when the evidence refutes them, whereas a probabilistic system must rely on the mathematics of its algorithm responding correctly to every situation; when something goes wrong, there may be no way to catch the error, sending a robot walking at full speed off the field, for example.

3 Contributions

This project has produced several key pieces of a cognitive localization system that each address an interesting facet of the problem of localization. First, as a proof of concept and a test of the basic ideas of this project, a new goalie system was created for the Northern Bites' code base. This system uses only visual information, with no sophisticated self-localization model, to allow the goalie to navigate the field and play effectively. Second, an object projection system was developed with dual purpose: to serve as an offline testing mechanism, replacing the physical camera and vision system, and also to use on the robot to make predictions about which objects may be seen from a certain field position. Finally, a visual memory system was produced that can store recently-seen objects in a way that would allow a cognitive system to use them effectively for modeling and predicting. Each of these three developments is discussed in depth in the following sections.

3.1 Proof-of-Concept: Northern Bites' Goalie

The goalie is a particularly interesting soccer agent because of its unique role in the game and its special field position. Most soccer players could visit any part of the field during a game when switching between offensive and defensive roles. The goalie, however, is an exception; it should always be located in the center of its own goal, facing the field so as to be prepared to block the opposing team's shots or clear the ball from the goal area. If it ever leaves this "home" position to chase the ball or becomes penalized, it is absolutely imperative that the robot returns home as soon as possible because the goal will otherwise

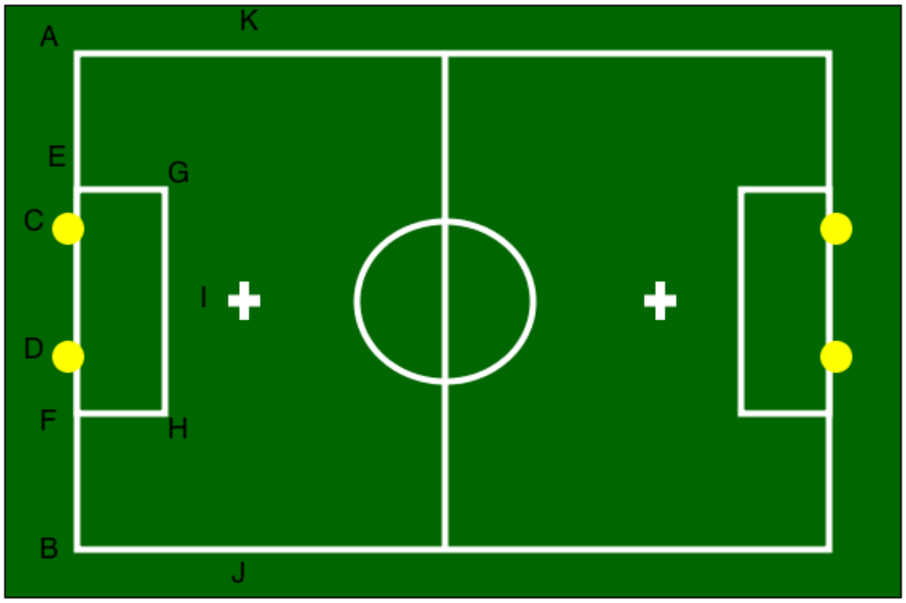

Figure 2: A view of the soccer field with significant goalie landmarks labeled as if the goalie is defending the left goal. The goalie's closest landmarks are the posts (C and D), the goal box L corners (G and H), and the goal box T corners (E and F). The outer L corners (A and B) and the field cross (I) are also significant. When penalized, the goalie returns to play at J or K.

be unguarded and vulnerable. In the SPL, no robot on the defending team other than the goalie can enter the goal box, so an absent goalie essentially means an open goal for the opponent [9].

The goalie's home position is equally interesting; it is an unusually landmark-rich location on the soccer field. With the field size increasing significantly for this year's competition, there are large areas of the field where few corners or goalposts are visible, but from the goalie area, many landmarks are immediately visible or easily found. These include the two goalposts, the two L-shaped goal box corners, and the two T-shaped goal box corners closest to the goalie, as well as the L-shaped field corners and the field cross within a reasonable distance. Each of these landmarks is labeled in a diagram of the soccer field in Figure 2. This large collection of usable landmarks means that the goalie is consistently given much more information about its location than other robots on the field.

The combination of the goalie's unique role and unique position means that there is much goalie-specific knowledge that can be brought to bear when having the player move around its area. Because the goalie's behavior is limited by its role in the game, it can make a very significant assumption about its location on the field: it is always very close to its own goal. Barring the exception of being penalized and removed from the game, the goalie's behaviors carefully keep the robot in a position to make saves and clear the ball from the goal area. While the visual goalie system takes this assumption as one of its key pieces of information, it is not so clear how it would be incorporated into a probabilistic localization filter.

In fact, this is one of the biggest weaknesses of using the standard localization system to inform goalie behaviors. A probabilistic filter makes no assumptions about the robot's location; this makes sense because a typical soccer agent could be at any location on the field. When provided with ambiguous landmark identifications from vision, it cannot simply throw out possible identities that may be obviously wrong when provided with some outside information. For example, if a corner is identified simply as a "goal box L," there are four possible concrete corners that could correspond to this identity, and a non-goalie agent could potentially be near any of them. In contrast, it is impossible for the goalie to see the opposite goal box's L-shaped corners, so two identities can immediately be eliminated, and with the additional assumption that the robot is facing the field, a definite identification for the corner can be made.

In contrast, if goalie is making decisions based on the traditional localization system and it considers positions at its own goal and the opposite goal, it may actually decide that it is located at the wrong goal and attempt to reposition itself. In this situation, the goalie could end up walking across the field to the other team's goal, thus creating many strategic problems for its team as described above. This problem in fact affects many SPL teams' goalies and is a strong argument for a system that recognizes that the goalie is special and uses related high-level positioning knowledge accordingly.

For this project, the method for incorporating this knowledge into the goalie robot's localization and navigation decisions was simple: the goalie was made to ignore any input from the particle filter. Rather than continuously maintaining any internal model of its environment or position, the goalie relies on direct visual perception of field landmarks to guide its behavioral decisions and movements. While this approach is atypical in the RoboCup domain, it actually follows in the long tradition of behavior-based robotics, a subfield of robotics that rejects AI's historical focus on knowledge and knowledge representation in favor of creating reactive, environment-driven behaviors [2]. A key tenet of behavior-based robotics is that "the world is its own best model," so internal representations of the environment are not necessary to effect intelligent behavior [5].

3.2 Principles of Goalie Functionality

The key to making such a low-level visual system work is carefully controlling the robot's behaviors so that the simplifying assumptions made during navigation are usually correct. Figure 3 is an overview of how the goalie's behavior system works. The goalie system, like all of the Northern Bites' behaviors, is implemented as an extended finite state machine, where the robot can be in only one state at a time and transitions between them based on the information received from the environment [13]. There are several points to notice with respect to controlling the robot's navigation, but the main idea is that the goalie must never engage in activities that could cause it to lose its place in the information-rich area near the goal and end up wandering around the field.

The central behavioral state, *watch*, is important because it enforces the restriction that the goalie is seldom moving and generally assumes that it is correctly positioned. This is in contrast to the field player robots, which spend much of their time repositioning to different field positions specified by the overarching behavior strategy. With the goalie, potentially disastrous movements must be minimized, meaning that its default strategy is to simply stand still and watch for the ball. However, this need to keep the goalie well-positioned in the goal is balanced by the need to have it act in the event that the ball is dangerously close to the goal. Hence the situations where the standing and waiting behavior can be overridden are as follows:

1. *Saving*: This is clearly one of the goalie's main responsibilities. When the ball is moving quickly toward the goal, the robot can decide to perform a squat save (see

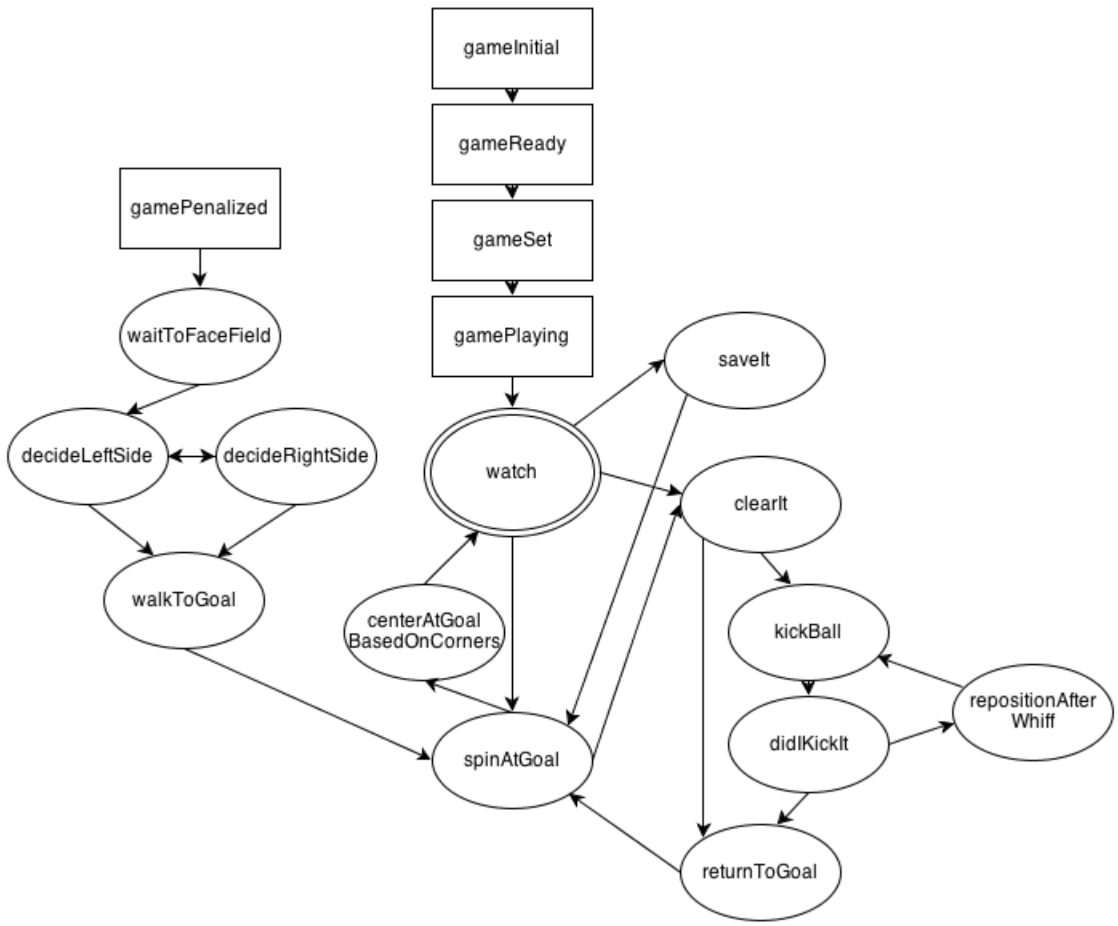

Figure 3: An overview of the goalie's complete behavioral system. Rectangular states are enforced by the SPL GameController and must be implemented by every robot. When the robot is told to enter *gamePlaying* or returns from *gamePenalized*, it can enter its goalie-specific behavioral states (in ellipses). Arrows represent transitions that can occur between states.

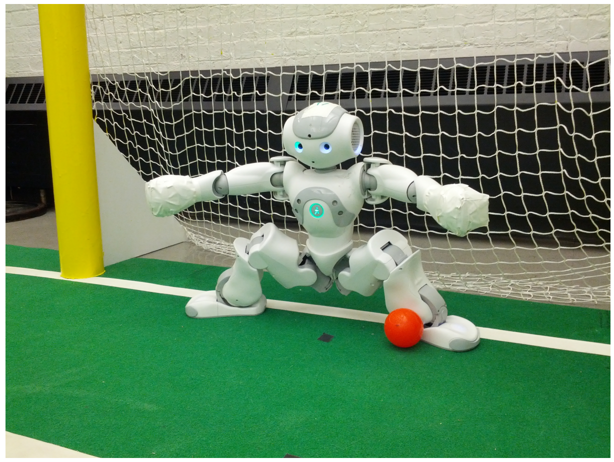

Figure 4: A Nao robot performing the Northern Bites' goalie save, which was developed by team member Dani McAvoy.

Figure 4). This movement will lead to the goalie repositioning itself once the save is complete and is particularly risky because it occasionally leads to the robot falling down. The need to keep the goalie standing and facing forward led to the choice of the squat save over diving saves, which are also available in the Northern Bites' code base. However, the chance that the goalie might prevent a goal with this save makes it worth the risk of confusing the robot's localization.

2. *Clearing*: Since no other robots on the defending team can enter the goal box, it is the goalie's responsibility to clear the ball if it is within the box. Thus, this is another situation where the chance of the goalie losing its home position is significant but that some action is necessary. The techniques used to minimize the risks involved with this action will be discussed in following sections.

3. *Extreme Confusion*: If the robot can recognize that it is clearly not facing the field as it should be, it may be worthwhile to have the robot attempt to reposition itself. This will only occur if the robot is extremely certain that it is facing in an incorrect direction—for example, it is directly facing one of its own goalposts. In this situation, having the robot attempt to spin to face the field is worth the risks of motion; otherwise, none of the goalie's other behaviors will be effective. Note that in this case, a cognitive goalie has a key advantage over a probabilistic robot: an extremely confused probabilistic system may have the robot wander in any direction looking for data, thinking that it could be anywhere on the field. A cognitive system would have a good idea of where to turn and look for enlightenment; for example, to place the visual field edge far away so as not to be looking off the field at nothing.

3.3 Safe Implementation of Goalie Navigation

The implementation of several of the goalie's states and transitions are highlighted below. While the details of various methods are interesting, there is one overarching idea: breaking the problem of the goalie's navigation into smaller sub-problems and solving each of these problems individually using the most relevant visual heuristic. This approach was inspired by the theory that humans often solve seemingly difficult tasks with extremely simple methods; for example, rather than subconsciously plotting the parabolic curves traced by a thrown ball, a baseball player utilizes a very basic eye-tracking method to position himself to catch it [11]. In the same way, the robot goalie is programmed to accomplish the complicated tasks of positioning, chasing, and saving by fulfilling simple sensory conditions.

- *spinAtGoal*: When the goalie is in the correct place on the field but may not be facing forward as it should be, it enters this state. One of the clearest visual indicators of whether the robot is looking toward the field is the visual field edge distance, which represents how far away the edge of the green field appears. Another important landmark that the goalie should see when facing forward is the field cross. Thus, in order to spin until it is facing forward, the robot looks straight ahead and spins in a circle until it either sees the cross in front of it or the field edge is far enough away to be the opposite edge of the green. This is a good example of a very basic visual heuristic that successfully positions the goalie toward the field so that it can watch for the ball.

- *clearIt*: The goalie's ball tracking information comes from a probabilistic filter, so approaching the ball is not one of the tasks that the robot accomplishes visually. However, one interesting problem when the goalie leaves the goal to chase the ball is that the robot often walks though the ball or whiffs a kick, so the ball moves slightly and the robot continues chasing it. Without checks, the goalie can wander very far from the goal because of these unintended ball movements—which completely ruins the robot's assumption that it is always in the goal area. A very simple solution prevents the goalie from wandering: using odometry to set a hard limit on how far away from its home position the robot can walk. When the robot's joint sensors determine that it has walked a predetermined distance, no matter how close the ball remains, the goalie will give up and return to the goal. This allows the goalie to at least move the ball far enough away from the goal for one of its teammates to continue clearing it and also ensures that the goalie never walks far enough away from its home position to get lost. While this rule is extremely simple, it makes the goalie much more effective by keeping it close to the goal.

- *centerAtGoalBasedOnCorners*: When the goalie is close to its home position but is not exactly where it should be, the two L-shaped goal box corners are well-placed to help the robot relocalize. Once the robot has completed some action that could interfere with its positioning, it looks for one of these corners and attempts to keep that corner in its image. Based on the robot's assumption that it is inside the goal box and facing forward, it is possible to know exactly which corner it is sensing. This situation is relatively rare on the RoboCup field, and it allows the robot to know its position with as much certainty as the visual observations allow. Thus, the robot can compute exactly how far away it is from home and move toward a better location, updating its trajectory as it goes based on continuing observation of the goal box corner. This method allows the robot to reliably return to its home position, given that the visual observations and estimates of the corner's distance, bearing, and orientation are accurate.

- *returnToGoal*: Any player that is penalized by the referee is taken out of the game for thirty seconds then placed on the sideline facing the field cross (see Figure 2), and this rule is no different for the goalie [9]. This is the one unavoidable situation where all of the assumptions about the goalie's typical position are no longer correct, so having the goalie walk back to the goal from this position is surprisingly difficult. Several methods have been tested to solve this problem, and each has some downsides. The first idea was to have the goalie pan its head to look for both goalposts and compute a trajectory that would end between the two. While this was somewhat successful, once the robot reaches the goal area and can no longer see both goalposts, its vision system often misidentifies the right and left posts, which means that finding a center point becomes increasingly difficult and the robot often misses the goal entirely. A second method tested was to have the goalie simply spin to face the goal at the penalty position then walk straight ahead. While this method also resulted in an acceptable initial trajectory, the Northern Bites' walk is not reliably straight; a robot that is attempting to walk straight actually walks in a curve. As a result, this method led the robot to a completely incorrect position on the opposite side of the field. The most successful method has been to have the robot track one of its goalposts visually and simply walk toward it. Although the robot does not end at the center of the goal using this method, it usually reaches a position that is close enough to its home to use the *centerAtGoalBasedOnCorners* method to correct itself.

3.4 Goalie Strengths and Weaknesses

Now that various aspects of its functionality have been described, it may be clear that the goalie system in its current version has both positive and negative aspects. A more thorough discussion of these strengths and weaknesses, including how some of the problems could be solved without compromising the successes, follows. It is important to remember, however, that the goalie system as presented is a proof of concept; it demonstrates that a probabilistic self-localization system is not necessary for a soccer agent to successfully participate in a game. However, it does so in the most straightforward way possible, resulting in a system that clearly lacks sophistication but raises many interesting issues.

One of the system's most obvious strengths is that it ensures that the goalie will reliably be positioned in the goal, so it can play its role in the game as its team members would expect. Many SPL goalies suffer from localization-related issues that lead them to move around the field, turn into the goal, or generally become lost, since they rely on the same localization system as the field players. The Northern Bites' goalie, in contrast, reliably stays in the goal area and is usually well-positioned to see the field and make saves. Since the goalie being is position is an important part of the team's strategy, this advantage is significant.

On the other hand, the goalie's behavioral limitations are also a weakness. The need to keep the robot safely near the goal results in a conservative chase threshold and no diving saves; a more confident goalie that could relocalize from more positions would be more aggressive and possibly prevent more goals. Additionally, since the goalie does not move unless it absolutely must, if it ever does get lost in a situation where it is outside the goal box or turned to face the wrong direction, it will generally not correct itself. That is, if something unexpected happens, such as the goalie running into another robot or falling down, and the goalie becomes badly positioned, it will believe that it is at its home location when it is not. This means that all of its behaviors will much less effective.

From these limitations, it is also clear that using a system similar to the current goalie, with no absolute position model, is not a feasible strategy for the other players on the field, although a similar system that incorporates the extensions discussed in following sections and generalizes the current functionality may become so. Field players must make many more decisions than the goalie in many different field areas; their actions cannot be divided into a straightforward set of solvable sub-tasks in the same way that that the goalie's can, so having an estimate for the robot's absolute position on the field is necessary. While this is not to say that some of the principles learned from the goalie prototype are not applicable—visual comparisons, simple heuristics, and behaviors that cater to localization are all valid ideas that field players could also implement—some extensions and improvements need to be made before an entire soccer team can function with a similar method of localization.

Essentially this relates back to the strengths and weaknesses of a cognitive system versus a probabilistic filter. The goalie system's simplicity cannot compete with the sophistication of a well-implemented particle or Kalman filter, but these more traditional techniques have textbook implementations and are well-defined. The experimental nature of cognitive localization means that prototypes such as the goalie's visual navigation system do not have the same performance guarantees and large bodies of previous research supporting them, but they have the potential to improve robot localization by considering what makes human localization so robust and successful. In other words, a probabilistic system is clearly a good choice for most field players at the moment, but both the potential for improvement to cognitive systems and the failures of traditional localization make the visual system a better choice for the goalie.

3.5 Visual Projection System

In order to expand the ideas from the goalie system into a more general localization method, some machinery is needed to allow the robot to compare its current view of the environment to an expected or known view in a meaningful way. This raises two questions: how can an expected view be created, and how can it be compared to what the robot is currently seeing? The second part of this project, a system to simulate the output of vision processing given the configuration of the world, addresses the first question. An example of the output of the current visual projection system is shown in Figure 5. Such a system actually has many uses; it was originally conceived of as a way to run tests offline without having to acquire images and run the vision system. However, when run online as an element of the robot's processing, it can provide an expectation with which the result of actual vision processing is compared.

To see how this could become an important element of a localization system, consider the following usage. Given a current estimate of the robot's position, the projected visual output could be computed. Then, if the localization estimate is close to the robot's actual location, the predicted scene will be similar to the actual scene from vision, and the differences between the two will provide information that can help to correct the position estimate. If there is little or no similarity between the two views, then the position estimate must be inaccurate; in this case, the system can take the appropriate steps to reset the localization system or disregard its estimate. This idea could even be incorporated as a check for the current probabilistic localization system; a comparison of the predicted vision with actual observations could help provide a layer of reasoning that helps control when random particles are injected or when the system should be completely reset.

The creation of projected views is also closely related to the second question of how the viewpoints could be compared, although it does less to address this question directly. If we have two viewpoints, it would be useful to determine the transformation from one viewpoint to the other, since this will provide concrete data about how the expected and

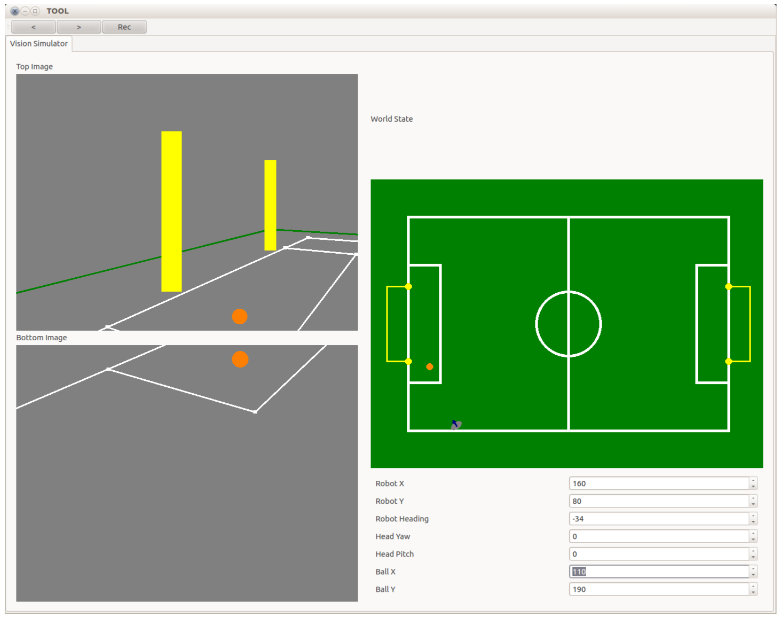

Figure 5: A screenshot of the visual projection system running offline. The right side of the screen allows the user to position the goalie and the ball, and the left side shows the robot's two projected images, which mimic the Nao's two cameras.

actual positions of the robot differ. As the following section will make clear, this process requires mathematical machinery very similar to that used to create the projections, which thus provide a good starting point for deciding how to compare two scenes.

On a side note, also consider the importance of simulation for development of vision-based systems such as localization. Testing is always problematic in the field of robotics because robots are designed to interact with the real world, which cannot be paused while a program is debugged, and merely watching a robot play soccer does not provide enough information about underlying systems' functionality to be directly helpful for dealing with errors and issues. In order to improve a system such as localization, offline testing is necessary to see exactly how the algorithm performs in certain situations.

Offline testing can be accomplished by running a robot around the field and recording a log of its sensor readings, then replaying these logs offline and feeding them into the localization system, but a more general way of simply producing approximate vision information without even needing a robot would be even more useful. Then, any field situation could be recreated without the time-consuming process of using a robot, and the amount of noise and interference in observations can even be adjusted to run increasingly stringent tests. This will produce quantifiable data about system improvements that may not be easily tracked in a real-world soccer game. In other words, the visual object projection system can also serve as the core of an important debugging tool for systems such as the memory system discussed later, traditional localization systems, or even behaviors.

3.6 Implementation of Visual Projections

The ultimate goal of the visual projection system is to produce the same output as the vision system, but without an actual image from the robot's camera. Instead of an image, the projection system takes as input a model of the physical world, which includes: the robot's global position (x, y, h), its head yaw, and the ball's position. These points are in the field coordinate system, where the origin is located at the bottom left corner of the field green with the x axis along the long end of the field, the y axis along the short end, and the z axis pointing up. Some simplifying assumptions about the world are made in order to not complicate the projection math unnecessarily. Namely, it is assumed that the robot's cameras are located at a fixed height, roll, and pitch. Ideally these values would be computed from the kinematics, taking into account every joint in the robot's body, but this would be unnecessarily complicated for the purposes of both simulation and prediction without significantly improving the output.

With these assumptions in place, the process of determining where a point in three-dimensional field space, would appear in the robot's image can be accomplished in two steps. First, the point in the field coordinate system needs to be transformed to a point in the camera coordinate system, where the origin is at the robot's camera, the z axis is along the camera's center of focus, the x axis points up from the camera, and the y axis points to the right. To accomplish this, using the robot's fixed head pitch p_f, current head yaw y, position (x, y, h), and the camera's fixed height h_f, a transformation matrix is composed from standard rotation and translation matrices [1]:

$$\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(p_f) & -sin(p_f) & 0 \\ 0 & sin(p_f) & cos(p_f) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} cos(y+h) & -sin(y+h) & 0 & 0 \\ sin(y+h) & cos(y+h) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right) \times \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -h_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the first matrix above takes into account the camera's pitch, the second the yaw—which must include both the robot's heading and the camera yaw—and the third the translation from the field's origin. Note that there is no need for a roll matrix because the roll is assumed to be zero; this matrix would simply be the four by four identity matrix.

Then, a point in homogeneous coordinates is created from the three-dimensional field coordinate and multiplied by the transformation matrix, from which the new three-dimensional coordinates for the field point in the camera coordinate system can be obtained. This sets up the second step, the actual projection of the point into the image plane. Given the focal length f_p of the robot's camera (in pixels), which is provided in the documentation, the pixel (x_i, y_i) can be produced from the three-dimensional camera-coordinate point (x, y, z) as follows:

$$x_i = (x/z) * f_p$$
$$y_i = (y/z) * f_p$$

Note that this image-plane point will be in the coordinate system where the origin is the center of the robot's image, but it can easily be transformed to typical image coordinates (where the origin is in the top left corner of the image) by adding half the image width to x_i and half the image height to y_i. Another detail that is important to the correctness of the projections is that points with negative z values in camera coordinates—that is, points behind the image plane—will be projected into the simulated image using this algorithm even though they cannot appear in the robot's camera image. Thus, a check for negative z values is critical so that objects will not be projected into the image from behind the robot.

In this way, any point in three-dimensional field space can be projected into the simulated robot image, which in turn means that the projection of any field object can be approximated. The simplest objects to consider are the corners, which can be thought of as single points $(x, y, 0)$ in three-dimensional field coordinates. Lines can be drawn by projecting the two corners that define their physical location into the image and connecting the two points, although there are some complications. There are three cases to consider:

1. If both corners are projected properly into the image plane, meaning that neither one has a negative z value in camera coordinates, then there is no issue and the line can be drawn between them normally.

2. If one corner is located behind the image plane, then projecting this corner into the image and connecting the line to it will produce unexpected results. Instead, the intersection of the line and the image plane in camera coordinates needs to be computed; the first line point will still be the correctly-projected corner, but the second will now be this intersection point.

3. If neither corner is in front of the image plane, then the line is not visible and can be disregarded.

The projection of the ball and goalposts can be estimated by projecting key points then computing the visual size of the object from its known real-world dimensions. For the ball, this means that its center point can be projected into the image and its visual radius calculated from its distance d and actual radius r using the focal distance f_c in centimeters and the conversion ratio of centimeters to pixels, c:

$$\text{visual radius} = (f_c/d) \times r \times c$$

A similar approach it taken for the posts, where the center of the base of the post is projected into the image and the visual height and width are calculated in the same way as the ball's radius. With the projection of the goalposts, the output of vision has been simulated successfully, so it can be placed into the same format as the result of actual visual processing and either compared to an observed scene or passed on to another system for testing.

3.7 On the Importance of Camera Calibration

One issue that the Northern Bites experienced this year is that the pitch, roll, and yaw of the Nao's cameras are actually slightly offset from what the robot's sensors report. That is, when a certain angle is reported to the vision system for use in its distance estimations, for example, that angle needs to be corrected on a per-robot basis. Each robot's camera, whether from manufacturing defects or rough falls, is slightly off from where the kinematics-based estimation system expects it to be, and every robot has a different amount of error.

In order to correct for these errors, the team had previously developed system where a robot is placed at a known location on the field and the expected visual lines can be projected into the image, using the kinematics-based transform that is used to make distance estimates in the vision system. This transform is similar to that described in the previous section but without the simplifying assumptions; it takes every robot joint into account. The result of the line projections for an uncalibrated robot can be seen in Figure 6.

As this figure demonstrates, the small unexpected discrepancies between angle sensor readings and the actual angle of the camera make a large difference in where the expected lines appear versus the actual field lines. Currently, the Northern Bites' platform allows

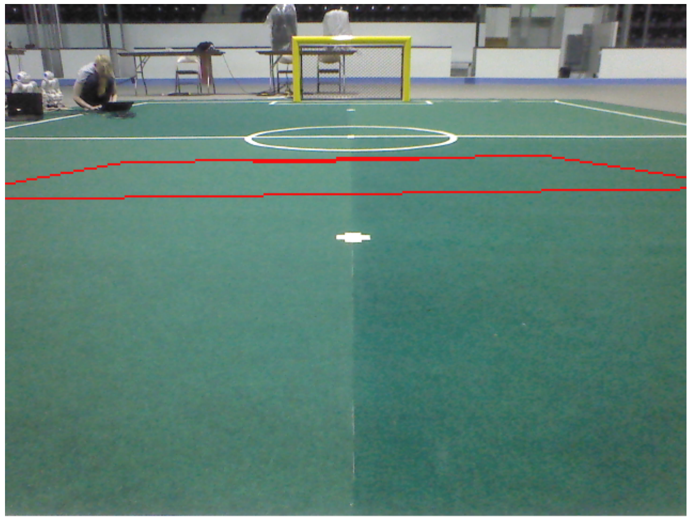

Figure 6: An image from the robot's camera with the expected visual lines, computed from the robot's position and joint angles, projected into the image in red. The discrepancy between the expected lines and the actual lines is the result of poor calibration.

constants to be added to the roll and pitch sensor angles to make up for the differences; these constants are obtained experimentally by projecting the lines as in Figure 6 then changing the corrections until the projections are well-matched to the actual image.

With this issue in mind, it is clear that camera calibration would have a large effect on any use of the visual projection system for localization. A small error in pitch or yaw would place the projected objects incorrectly within the image, meaning that any comparison to actual vision results would produce misleading information. This problem brings into focus some of the frustrations related to working with physical robots; while the projection system itself may be sound, this minor physical defect will have a large effect on how well it actually performs.

3.8 Visual Memory

As pointed out previously, memory as opposed to state as history is a key part of a cognitive system, and memory for recent visual observations could additionally be helpful to many different parts of the Northern Bites' soccer platform, including vision itself and behaviors. It could particularly improve the vision-based goalie system discussed above. The basic data structures of a visual memory system have been implemented, although it is left to future work to utilize the memory designed in this project across the various systems that could use it. The goal of this improvement is to provide the RoboCup robots with something akin to human working memory, which people utilize when they need to briefly remember and process information that is no longer provided by perception but is still needed to complete a given task [3].

Two ideas about management of stored information from cognitive theory of memory were used to inspire the memory system's functionality: trace decay, meaning that memory will "spontaneously decay unless refreshed," and interference, or the idea that new information will cause old information to be forgotten [3]. In addition, PLAN's focus on stored and abstracted scenes has been incorporated into the way that the memory system stores its information.

The memory system is, at its core, a ring buffer of scenes; it has a hard limit on how many scenes can be stored at any given time, so new data will by default overwrite old scenes once the buffer is full. The amount of time that can be stored in the buffer is a parameter that represents the effect of interference and can be set based on estimates from cognitive theory or empirical results. Every scene in the buffer contains an abstracted representation of each of the objects it contains. For example, the ball is represented by its center point in the image, its visual radius, and its distance and bearing estimates; this information attempts to capture all of the useful visuo-spatial information about a given object without keeping too many details.

In addition to the scene organization, each object is explicitly continued through time by being part of a linked list of similar objects in previous and subsequent scenes. That is, if a ball is visible in a scene, it will be connected to the ball from the previous scene and that in the next scene, meaning that the memory system can be traversed in two distinct ways. One is per-object: if a system is interested in how the ball moves over a series of frames, it can follow the ball's temporal list and gain that information. The other is per-scene, meaning that spatial relationships between objects can be traced by traversing a subset of the scene buffer.

Trace decay comes into play at the object level; each observation has a timestamp, and the system will remove any record of observations of an object that are older than a certain threshold. Since different objects can have different thresholds, some objects will be tracked more extensively than others, which is appropriate because the system may need to extract more information about the ball than about field lines, for example.

The memory system overall is designed to support cognitive reasoning—by the goalie system or any other system that needs to incorporate higher-level reasoning about visual and spatial locations of objects. Patterns that can only be seen over a series of frames currently cannot be detected by any system in the Northern Bites' platform because no history other than the current state is preserved. The information that this visual memory system can provide will likely prove useful to soccer playing in general, and it is certainly necessary for further development of cognitive spatial systems.

4 Future Work

This paper has presented several different elements that make steps toward a cognitive localization system for robotic soccer agents. The next part of this project would be to take the progress and insights from the pieces developed here and put them together to create an effective general localization system. The goalie's visual heuristics and localization-friendly behavior, a visual projection scheme for comparing expected and actual scenes, and memory for recent observations would all play a role in a more fully-developed localization system that could eventually be generalized to all players on the field.

In addition to the larger task of putting the pieces from this project together, the goalie, visual projections, and memory could also all be improved individually. In terms of the goalie, many improvements could be made to make it a more effective member of the team while still adhering to the principles that underlie its current functionality. For example, since returning to the goal from penalty is such a difficult problem, a better solution could be found to steer the goalie in the right direction based on a combination of field landmarks rather than simply the goalposts. Also, the idea of using entire scenes rather than individual landmarks to create visual heuristics is an interesting prospect that this project has not yet been able to explore.

While the visual projection system is complete and functional, it too could be improved in order to both make better predictions and simulate actual data better for testing. More sophisticated ways of projecting three-dimensional objects like the goalposts could be explored; while the current system makes reasonable estimates, more accurate goalpost projections, for example, would ultimately be helpful for both usages. The most useful extension that could be made for testing would be to add a controllable level of noise to the projections so that they reflect the imperfect nature of the robots' sensors and the vision system better; it would be worthwhile to see how systems being tested by the simulated vision output respond to varying levels of interference. If the projections are to be used on the robot, improvements to the camera calibration system may also be necessary—an automatic calibration system that can account for more degrees of freedom would be ideal.

The visual memory system has the most room for improvement; its basic functionality has been developed, but the ways in which it could be employed in the Northern Bites code base have yet to be explored. It could aid in vision processing, for example, by pointing out where an object was in the last frame so that processing could start in the same location, thus saving needless searching through the image. Visual identifications could also be improved because more information about the context of each observation will be preserved, meaning that localization would be passed better observation information. Of course, it could also be used to improve goalie functionality because of the increase in the amount of visual information available and the ability for high-level reasoning about sequences of scenes. All in all, there are many ideas that build upon the developments discussed in this paper and that should be explored more thoroughly.

5 Conclusions

While this project has been developed in several complementary parts, there are some overarching conclusions that can be taken away from looking at the various pieces as a whole. One of the most important is that a cognitive, non-probabilistic localization system is feasible in the RoboCup setting. Between the proof-of-concept in the simple goalie system and the potential improvements that would be gained by incorporating visual memory and expectations from visual projections, the different parts of this project point toward the viability of a cognitive localization system for soccer robots. This is an interesting conclusion from the point of view of the debate whether theories about human cognition can inspire or improve AI; the possibility that ideas from human cognitive maps could lead to an alternative method for robot localization is clearly a point in favor of considering human processing when attempting to solve problems in the domain of robotics.

Nonetheless, this project also highlights some of the reasons that mathematical approaches dominate robotics at this time. For one, they do work well in situations such as RoboCup soccer, where the environment is designed to give robots an advantage and is well-known. Additionally, when creating a probabilistic filter for localization, there are standard implementations and formulas that specify how different algorithms should work, meaning that these methods are not highly experimental and will likely function as desired if implemented correctly. Cognitive techniques provide no such guarantees, and there are no clear guidelines for programming a cognitive localization system—which of course makes this an interesting question for research, but difficult to put into place in a soccer code base. These difficulties reflect back onto the cognitive theories utilized in this project; while theoretical work clearly provides interesting and useful ideas, implementation details can completely make or break a cognitive system and provide insight as to where the theory fails. In this case, implementing a cognitive system on a robot has demonstrated that the underlying theory still needs to be fleshed out and expanded, particularly with concrete details that would allow a physical robotic system to actually run based on cognitive principles.

Furthermore, the experience of attempting to apply cognitive theories to RoboCup strongly supports the importance of robotics to AI in general. Working with robots is simultaneously the best and worst thing that a researcher can do when testing a new theory; the challenge of producing a system that interacts with the real world helps push ideas in interesting new directions, but often at a price. Robots are infuriatingly difficult to work with because problems completely unrelated to the question at hand can completely ruin an experiment and drain hours of effort away from academic concerns. At the same time, there is simply no better way to attempt to push AI toward human-level interaction with the world—without sensors and actuators that have to cope with the messy confusion that is reality, AI programs would simply solve toy problems and not progress toward the more general capabilities of an intelligent real-life agent.

Due to some of these frustrations inherent in robotics, while this project has certainly demonstrated reasons to move forward with a cognitive localization system, it also clearly shows why purely probabilistic approaches are clear choices for RoboCup teams at the current time. Cognitive methods, however, remain a promising option that could improve robotic function in complicated real-world environments—and indeed may be necessary for robots to cope with situations that are not engineered to recompense for their weaknesses. Thus, with a view toward the world beyond RoboCup, it has been a valuable exercise to attempt to employ cognitive techniques on the soccer field.

References

[1] Thomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-Time Rendering*. A K Peters, Ltd., Wellesley, Massachusetts, third edition, 2008.

[2] Ronald C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, Massachusetts, 1998.

[3] Alan Baddeley. *Working Memory*. Oxford University Press, 1986.

[4] Samuel Barrett, Katie Genter, Yuchen He, Todd Hester, Piyush Khandelwal, Jacob Menashe, and Peter Stone. Ut austin villa 2012: Standard platform league world champions. 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.

[5] Rodney A Brooks. Elephants don't play chess. *Robotics and autonomous systems*, 6(1):3–15, 1990.

[6] Maria Belen Canadas. Cooperative ekf localization. 2005.

[7] Eric Chown and Byron Boots. Learning cognitive maps: Finding useful structure in an uncertain world. In *Robotics and Cognitive Approaches to Spatial Mapping*, pages 215–236. Springer, 2008.

[8] Eric Chown, Stephen Kaplan, and David Kortenkamp. Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995.

[9] RoboCup Technical Committee. Robocup standard platform league (nao) rule book. `http://www.tzi.de/spl/pub/Website/Downloads/Rules2013.pdf`. Accessed May 6, 2010.

[10] The Robocup Federation. Objective. `http://www.robocup.org/about-robocup/objective/`. Accessed May 6, 2010.

[11] Gerd Gigerenzer. *Gut feelings: The intelligence of the unconscious*. Viking Adult, 2007.

[12] J. Hawkins and S. Blakeslee. *On intelligence*. St. Martin's Griffin, 2005.

[13] Tucker Hermans, Johannes Strom, George Slavov, Jack Morrison, Andrew Lawrence, Elise Krob, and Eric Chown. Northern bites 2009 team report. Technical report, Technical report, Department of Compute Science, Bowdoin College, 2009.

[14] Leopoldo Jetto, Sauro Longhi, and Giuseppe Venturini. Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots. *Robotics and Automation, IEEE Transactions on*, 15(2):219–229, 1999.

[15] Thomas Röfer, Tim Laue, Judith Müller, Michel Bartsch, Malte Jonas Batram, Arne Böckmann, Nico Lehmann, Florian Maa, Thomas Münder, Marcel Steinbeck, Andreas Stolpmann, Simon Taddiken, Robin Wieschendorf, and Danny Zitzmann. B-human team report and code release 2012, 2012. Only available online: `http://www.b-human.de/wp-content/uploads/2012/11/CodeRelease2012.pdf`.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* The MIT Press, Cambridge, Massachusetts, 2006.

[17] Edward C Tolman. Cognitive maps in rats and men. *Image and environment: cognitive mapping and spatial behavior*, pages 27–50, 1973.