A Decision-Making and Actions Framework for Ball Carriers in American Football

Danny Jugan and **Dewan T. Ahmed**
University of North Carolina at Charlotte
9201 University Blvd.
Charlotte, NC

Abstract

Instructing intelligent agents in team-based, multi-agent environments to respond to dynamic events is a lengthy and expensive undertaking. In this paper, we present a framework for modeling the decisions and behaviors of ball carriers in American Football using the Axis Football Simulator. While offensive strategies in football employ the use of prescribed plays with specific spatio-temporal goals, players must also be able to intelligently respond to the conditions created by their opponent. We utilize a two-part substate framework for ball carriers to advance downfield while avoiding defenders. Unlike existing football simulations that employ variations on professional football rules and regulations, our method is demonstrated in a realistic football simulation and produces results that are consistent with actual competitions.

Introduction

When creating simulations or games with the intent to teach, it is necessary to provide the user with an environment that closely matches the real-world counterpart [1]. This includes not only an appropriate visual representation, but also an accurate resemblance of the decisions and actions of the self-governing agents in the environment. In sports simulations, additional challenges exist in creating a realistic environment due to the added strategy, coordination, and dynamic nature that is inherent to sporting competitions.

While there are several published works related to controlling players or overall team decisions — such as coaching responsibilities — in sports simulations, there does not yet exist a framework for managing the behaviors of players in American Football (football). The sports-based methodologies that do exist are either too generic or not applicable to football and are therefore unable to be utilized. Additionally, much of the published research pertaining to football exists in simulations that do not accurately reflect the rules, regulations, or conditions for professional football.

The goal of this paper is to present a framework for modeling the decisions and actions of an offensive ball carrier. The methods used in the framework will be implemented using the Axis Football Simulation (Axis) — a 3D American Football simulation created with Unity. Since the simulation can be used as a tool to train football coaches and players, the overall objective of the agents, under direction of various algorithms, is to collectively provide an environment that closely matches the decisions, actions, and capabilities of players in an actual football competition. Therefore, in an effort to maintain the validity of the simulation, behaviors that extend beyond normal human ability (e.g., processing information faster than possible, using data that an actual player would not have access to, and so forth) will be avoided. Additionally, Axis' rules and regulations (e.g., field size, player capabilities, active participant numbers, and so forth) are intentionally designed to be consistent with actual football competitions.

While Axis is running, the user will always control a single character, leaving 21 other agents to be controlled by intelligent scripts. Agents can be categorized into one of a finite number of states based off of their current goal(s). Those goals are established by combining the prescribed instructions of the selected play and dynamic adjustments made in response to the changing conditions of the environment (i.e., the states and positions of the surrounding players and the location and possession of the ball). While on offense, an agent will fall into one of four goal-oriented states: RUNNING, THROWING, RECEIVING, or BLOCKING. On the defensive side, agents will also be grouped into one of four states: ZONE COVERAGE, MAN COVERAGE, BLITZING, or TACKLING. While this paper will focus only on the RUNNING framework, we believe the combination of the individual methodologies will produce behaviors that closely resemble the actions of professional football players at both the individual and team level.

The RUNNING state will function independently as a hierarchical state machine divided between positions of behind and in front of the imaginary line on the field where the play begins (i.e., line of scrimmage). When the player receives the ball behind the line of scrimmage, a series of raycasts will be performed in the surrounding area in order to determine the best place to run. If the runner crosses the line of scrimmage, they switch to a different substate that detects and avoids nearby defenders. We proposed that this two-part framework for controlling offensive ball carriers provides a realistic simulation of decisions and behaviors taken by actual players in football competitions.

Related Work

In demonstrating proposed models or methodologies for a particular set of agents in a sports environment, it is necessary to have a simulation in which they can be applied. Rush 2008, a research extension of Rush 2005, simulates play in an eight player variant of American football (football) [7]. While it was originally developed as a platform for evaluating game-playing agents, several researchers have utilized the simulation to produce papers related to modeling or developing learning strategies.

Laviers *et al.* presented an approach for online strategy recognition [6]. Using information about the defense's intent (i.e., play history aggregated from spatio-temporal traces of player movements), their system evaluates the competitive advantage of executing a play switch based on the potential of other plays to improve the yardage gained and the similarity of the candidate plays to the current play. A play switch, unlike an audible that changes the play before it starts, makes adjustments to the prescribed movements and responsibilities of selected agents *after the play has started*. Their play switch selection mechanism outperforms both the built-in Rush offense and a greedy yardage-based switching strategy, increasing yardage while avoiding mis-coordinations induced by the greedy strategy during the transition from the old play to the new one.

Laviers and Sukthankar later created a framework for identifying key player agents in Rush 2008 [5]. They discovered that in football, like many other multi-agent games, the actions of all the agents are not equally crucial to game-play success. By automatically identifying key players from historical game play, the search space can be focused on player groupings that have the largest impact on yardage gains in a particular formation. Within the Rush football simulator, they observed that each play relied on the success of different subgroups — as defined by the formation — to gain yardage and ultimately touchdowns. They devised a method to automatically identify these subgroups based on the mutual information between the offensive player, defensive blocker, and ball location and the observed ball work flow. Laviers and Sukthankar concluded that they can identify and reuse coordination patterns to focus the search over the space of multi-agent policies, without exhaustively searching the set partition of player subgroups.

Li *et al.* took a different approach and outlined a system that analyzed preprocessed video footage of human behavior in a college football game and used it to construct a series of hierarchal skills[7]. The learned skills incorporated temporal constraints and provided for a variety of coordinated behavior among the players. They used the ICARUS architecture as the framework for play observation, execution, and learning, which is an instance of a unified cognitive architecture [8]. Since reasoning about the environment is a principal task for intelligent agents, ICARUS supplies agents with a combination of low-level perceptual information (i.e., attributes of a single environmental object) and higher-level beliefs(i.e., relations among objects). After inferring that set of beliefs about its environment, ICARUS next evaluates its skill knowledge to determine which actions to take in the environment. For this, the architecture uses a goal memory, which stores goals that the agent wants to achieve.

The method Li *et al.* used for acquiring the constructed hierarchal skills was divided into three steps. First, the system observes the entire video-based perceptual sequence on actors achieving the intended goal and infers beliefs about each state. Next, the agent explains how the goal was achieved using existing knowledge. Finally, the algorithm constructs the needed skills along with any supporting knowledge, such as specialized start conditions, based on the explanation generated in the previous step. Those skills were tested using Rush 2008, and although the level of precision in ICARUS' control was found to need improvement, the results suggested that their method was a viable and efficient approach to acquiring the complex and structured behaviors required for realistic agents in modern games.

Figure 1: Sample player formation from the Rush 2008 football simulator.

While all of these studies produce improvements to models or frameworks associated with football strategies, they make no mention of the specific decisions that the individual players use while the simulation is running. Additionally, as shown in figure 1, a main problem exists that the simulation in which their strategical improvements are found does not accurately represent a football environment (e.g., Rush 2008 uses eight players instead of eleven). Removing three players from each team has a drastic effect on not only the strategic planning process, but also on the dynamic spatio-temporal aspects of live play. As stated in a sports-based game simulation research report, the development of a program for playing a sports game presents a problem in that the simulation of the game itself must be done in such a way so that the physical interaction of the game is represented accurately [1]. Some have addressed this problem not by representing the physical performance (i.e., decisions and actions) of the player, but rather by using statistical data to determine the success or failure of a play based on actual National Football League statistics.

Gonzalez and Gross use this approach in their coaching football simulation. The objective of the program — which provides two basic decisions to make: what offensive play to execute (if on offense), and what defensive formation to use (if on defense) — is to make the best selection possible based on the rules of the game, on a priori knowledge about strategy, and on learning from the opponent's play selection [1]. When the program is faced with selecting an offensive play from a set of defined plays, it will first ascertain the conditions — taking into account the field position, down, yards to go for a first down, and the score. Those game conditions will then be compared with the ideal conditions defined for every play, and the difference in each element will be designated as a *delta value*. The set of ideal conditions is part of the a priori knowledge derived from experts and the closeness of the current conditions to the ideal conditions defined for each play is the main influence on the play selection. The statistical history used by the program, which describe how successful each play has been, also has a major impact on play selection. On the other side, the program chooses the most appropriate defensive formation by attempting to guess what the opponent will try to do under the present game circumstances. It uses the historical database to formulate the opponent's tendencies by employing a heuristic function that determines which offensive play the opponent is most likely to select. The program then selects the defensive formation that is most effective against that play. While this type of intelligent formation control and learning is an important aspect of football simulations, it relates only to the play selection component and does not address the specific actions of the individual players.

Related research has also been conducted in the area of complex multi-agent probabilistic actions where *complex* means the actions contain many components that typically occur in a partially ordered temporal relation and *probabilistic* refers to the uncertain nature of both the model and data. Essentially, the work surrounds the idea of evaluating whether an observed set of actions constitutes a particular dynamic event. Intille and Bobick presented a model for representing and recognizing complex multi-agent probabilistic actions in football [3]. Using prior work in tracking football players from video, they first define a *temporal structure description* of the global behavior (i.e., a football play) [2]. The basic elements of this structure represent individual, local goals or events that must be detected. For each basic element of the temporal structure, they define a visual network that detects the occurrence of the individual goal or event at a given time accounting for uncertain information. Temporal analysis functions are then defined to evaluate the validity of the set of temporal relationships (e.g., does one action typically precede another). Finally, a large multi-agent belief network is automatically constructed reflecting the temporal structure of the action. Uncertain evidence of temporal relationships between goals is sufficient to cause the play detector's likelihood value to quickly rise above the other plays shortly after the play action begins at frame 90.

While Intille and Bobick's work can detect — with relative uncertainty — the goals and perhaps future actions of the offensive agents, doing so requires knowledge of the offense's playbook (i.e., complete set of available plays the offense can run), something that is not only dynamic, but also secretive. Without the set of predefined temporal-action relationships, the defensive agents would be forced to build the dataset from observed plays during a game, and would then only be able to properly identify — again with uncertainty — plays that it has already observed. Additionally, as the number of plays the offense has the ability to run increases, the number of options the defensive agents must consider — and ultimately the complexity of the decision — also increases. Still, their work provides a building block that could serve as a framework for instructing individual defensive agents attempting to determine the offense's intent.

Stracuzzi *et al.* build on that work by proposing an application of transfer from observing raw video of college football to control in a simulated environment. The goal is to apply knowledge acquired in the context of one task to a second task in the hopes of reducing the overhead associated with training in the second task [10]. In the initial task, the system must lean to recognize plays, including the patterns run by individual players. The second task requires the system to execute and improve upon the plays observed in the source in a simulated football environment (Rush 2008). Their transfer system consists of three distinct parts. The first part, which corresponds to the source-learning task, takes the raw video along with labeled examples as input and applies statistical machine-learning techniques to distinguish among the individual players on the field and to recognize the activities of each player. In the second part, the system maps recognition knowledge acquired in the source into the procedural knowledge required by the target. For this, the system uses the ICARUS architecture discussed above. In the third part, the system uses ICARUS to control players during simulation and adds a heuristic search mechanism to support adaptation of the learned plays to the simulated environment. Their work provides a clear framework for action recognition in a complex environment, transfer of action recognition into procedural knowledge, and adaptation of the constructed procedures to a new environment. Overall, research in transfer of learning has great potential to affect the manner in which we train intelligent agents. However, a major limitation to their approach is that it utilizes, as the source, only passing plays, so the offense has no ability for an application of transfer to running plays. Additionally, the implementation of their strategy is done using the Rush 2008 simulator. We previously mentioned the rule variations between American football and Rush 2008 with regard to the number of active players, but another important distinction between the two is the size of the field. Rush 2008 uses a wider field than regulation, allowing for additional open spaces with which to complete passes. This naturally has an effect on the success of those plays. Finally, the transfer system controls only the offensive players, while the simulator is left to control the defense. This means that the defensive strategies utilized by the simulator may be very different from that of an actual player.

A last piece of related work focuses on the pursuit of human-level artificial intelligence and its application to interactive computer games. Laird and van Lent describe human-level AI as being able to seamlessly integrate all the human-level capabilities: real-time response, robustness, autonomous intelligent interaction with their environment, planning, communication with natural language, commonsense reasoning, and learning [4]. Additionally, they pro-pose that the increasing realism in the graphic presentation of the virtual worlds has fueled — and even necessitated — the corresponding increase in more realistic AI. Indeed, their work validates our claim that increasing the visual realism in sports simulations (i.e., 2D pixel graphics in Rush 2008 to 3D modeled players in Axis), there has to be an accompanying increase in the sophistication and capabilities of the intelligent agents present in the environment.

Methodology

In American Football (football), the goal of the ball carrier is to advance as far down field as possible — with the ultimate goal of reaching the end zone. Our system places the ball carrier in a hierarchal RUNNING state, which happens whenever an offensive player receives the ball (e.g., handoff, completed pass, or recovered fumble), the quarterback crosses the line of scrimmage with the ball, or a defensive player gains possession of the ball (i.e., a turnover).

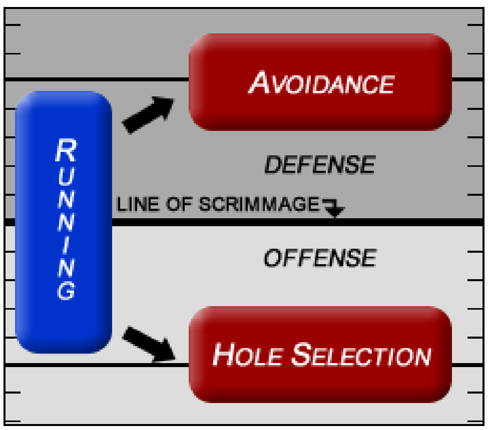

Figure 2: Ball carriers in the *Running* state will be placed in the *Hole Selection* substate while they are behind the line of scrimmage and will transition to the *Avoidance* substate once they cross it.

The methodology for advancing down field under the RUNNING state is divided into two substates: HOLE SELECTION and AVOIDANCE. As shown in figure 2, the substate is determined by the position of the player in relation to the line of scrimmage. If an offensive player receives the ball behind the line of scrimmage, they enter the HOLE SELECTION substate. All other conditions — including the player crossing the line of scrimmage while in the HOLE SELECTION substate — will result in the player entering the AVOIDANCE substate. While the focus of this paper is on the running aspect of the simiulation, it is also important to note that a player who catches a pass will immediately enter the AVOIDANCE substate regardless of their position in relation to the line of scrimmage.

Hole Selection

The goal of the HOLE SELECTION substate is to find the best hole (i.e., gap between players) through which to run. It is important to note that while designed running plays have prescribed paths that runners are instructed to follow, it is impossible to predict the defensive player's actions. Therefore, in order to maintain acceptable levels of intelligence, actual paths — and ultimately holes — must be determined dynamically. When an offensive player receives the ball behind the line of scrimmage, they determine the best hole through which to run by factoring the size of the available holes with the player's proximity to those holes. Holes themselves are determined by decomposing a rectangular area of the field surrounding the blockers at the line of scrimmage. As shown in figure 3, the raycasts are spaced at a static interval slightly smaller than the approximate shoulder width of an average player. We will discuss the effectiveness of various raycast spacing strategies in subsequent sections.

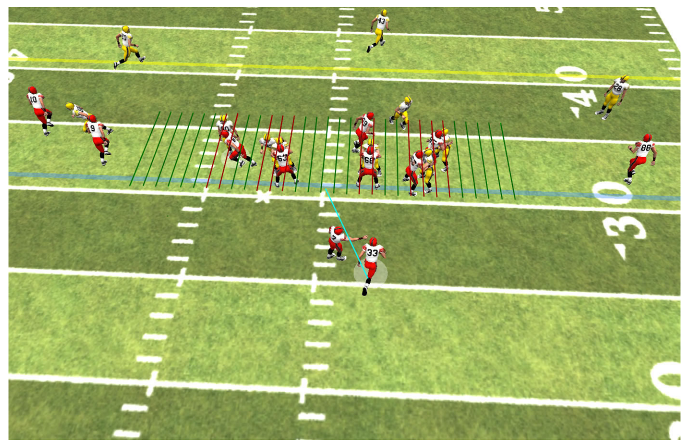

Figure 3: The runner (number 33) selects the best hole through which to run. Red lines indicate raycasts that collide with at least one player.

Once the raycasts are executed, the area is decomposed into a single dimension Boolean structure reflecting whether or not the casts collided with any player. The structure is then transformed to reflect the number of consecutive empty positions at each hole. The player's position is normalized along the width of the decomposition area to determine the index of the structure closest to the player. The system then generates a value for each hole within five positions to the left and right of the player's index. The value is calculated by taking the size of a hole and subtracting from it the distance in positions from the player. The largest resulting value is chosen as the hole through which to run, and the player selects the midpoint of that hole at the line of scrimmage as the goal position. The entire algorithm for the process can be seen in algorithm 1. Once the runner reaches the goal position, it transitions to the AVOIDANCE substate.

Three important notes are worth mentioning with respect to the methodology used during the HOLE SELECTION substate. First, when the initial raycasts are made to decompose the area around the line of scrimmage, collisions with either offensive or defensive players will flag that position as non-empty. Logical arguments can be made that only defen-

Algorithm 1 Runner Hole Selection

```
function FindHole(decSize, spacing, vision)
    startPos ← runnerPos.x − (spacing ∗ decSize/2)
    holes[ ] ← new [decSize]
    for i ← 0 to decSize − 1 do
        castLoc ← startPos + (i ∗ spacing)
        if Raycast(castLoc) then
            holes[i] ← 0
        else
            holes[i] ← 1
    for i ← 0 to decSize − 1 do
        if holes[i] = 1 then
            j ← 1
            while j+1 < decSize and holes[j+1] ≠ 0 do
                j ← j + 1
            v ← j − i + 1
            for j ← i to i + v do
                holes[j] ← v
            i ← v + i
    index ← (player.x − startPos) / spacing) + 1
    start ← max (index − vision, 0)
    end ← min (index + vision, decSize)
    bestPos ← 0, bestVal ← −1
    for i ← start to end do
        v ← (vision− abs(index − i)) + holes[i]
        if i > bestVal then
            bestVale ← v
            bestPos ← i
    if bestPos < index then
        return startPos + (bestPos ∗ spacing) − ((holes[bestPos]/2) ∗ spacing)
    else
        return startPos + (bestPos ∗ spacing) + ((holes[bestPos]/2) ∗ spacing)
```

sive players pose a threat to the runner, and that offensive players should be ignored during decomposition. When attempting to determine the best approach to selecting a hole for the runner, we ran tests with casts that both collided with and ignored offensive players. However, ignoring offensive players provides information to the runner beyond what they have the ability to observe. Because the runner is behind the offensive linemen at the time that the raycasts are made, the line of sight to potential defenders directly in front of those linemen will be obscured. Therefore, it is unrealistic to expect the runner to be able to accurately see the positions of those players. Additionally, ignoring offensive player collisions with the raycasts caused the runner to unnecessarily collide with a lot of their blockers, reducing their ability to advance downfield.

The second important distinction relates to the lateral distance with which the runner is able to view when determining which hole to select. The average human has a peripheral vision extending 120 degrees, so holes beyond five positions from the runner were beyond their perceived vision and were ruled out as unrealistic choices. On the other hand, reducing the lateral visibility distance to four or fewer positions caused the runner to miss holes of a larger size that were perceived to be in their field of view. Ultimately, extending five raycast positions was found to produce results that were the most consistent with the perceived vision limits of the runner.

Finally, it is also important to note that the decomposition and hole selection occurs only once at the time of the handoff. Our initial approach utilized a continuous check (i.e., the runner would decompose and select the best hole at each game step), but this produced erratic behavior from the runner that was inconsistent with actual player behaviors. Furthermore, we believe that continuously decomposing the environment using our algorithm extends beyond normal human processing and reaction abilities and places unnecessary stress on the simulation.

Avoidance

The goal of the AVOIDANCE substate is to advance as far downfield as possible while avoiding defending players. The runner employs the use of a 120 degree field-of-view reaction radius for the basis of its running direction. The bounds of the cone-shaped radius are determined by the dot product of the runner's normalized forward vector and the normalized vector to potential tacklers. Defenders outside of the cone — including those that are behind the runner — will be ignored under the premise that the runner would not be able to see them. If the runner's reaction radius yields no threats, the player is instructed to run straight forward towards the end zone at their current lateral position on the field.

Figure 4: The dashed red line coming from the runner (number 82) identifies the closest defender, and the blue arrow shows the angle he will take in an attempt to avoid being tackled.

If there are defenders inside the radius, one of two scenarios can occur. First, if there is only one defender or all of the defenders are on only one side of the runner, the runner will attempt to avoid the threat by taking an angle away from the closest defender. This is shown in figure 4. Second, if there are threats on both the left and right side of the runner, the player concludes that collision is unavoidable and attempts to gain as much yardage as possible prior to being tackled. This, like when there are no threats, is achieved by running straight forward along the runner's current lateral position.

Experimentation

For the two substates of RUNNING, a series of tests were run to determine the specific values that yielded the highest results for each of the individual methodologies. For all of the tests, a series of 100 plays were run with all other factors (defensive play calls, agent behavior, and so forth) remaining constant.

Hole Selection

The first set of tests was designed to determine the most accurate and efficient way to decompose the area where the ball carrier is attempting to run. Decomposition refers to the structuring, at the variable level, of the physical objects present in the simulation. Our goal was to identify the specific locations of the players — and ultimately the spaces between them — while minimizing the processing requirements on the simulation.

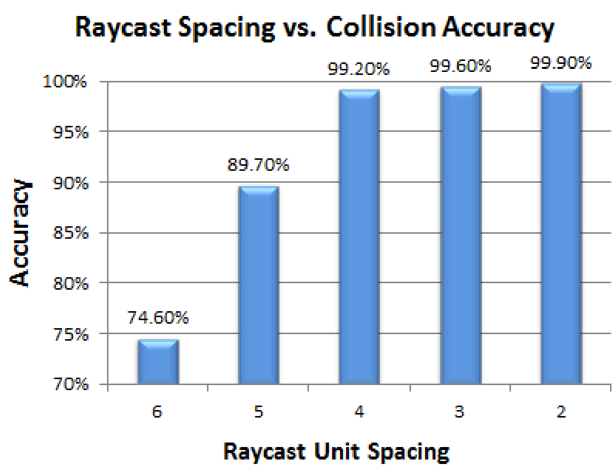

Figure 5: The results of 100 test plays at a variety of raycast spacing values. The width of one unit is approximately one-fourth the width of a player.

Our first goal was to determine the maximum lateral spacing between each raycast that would produce an appropriate environmental representation. Figure 5 shows the accuracy of a variety of raycast quantities within the same decomposition area in determining the existence of a player at that position. Each of the raycast spacing distances was tested by executing 100 plays and the accuracy of the raycasts in identifying the actual location of the players was recorded. For the purpose of testing, an arbitrary unit spacing value was defined with the value of a single unit approximating one-fourth the width of an in-game player. The results show

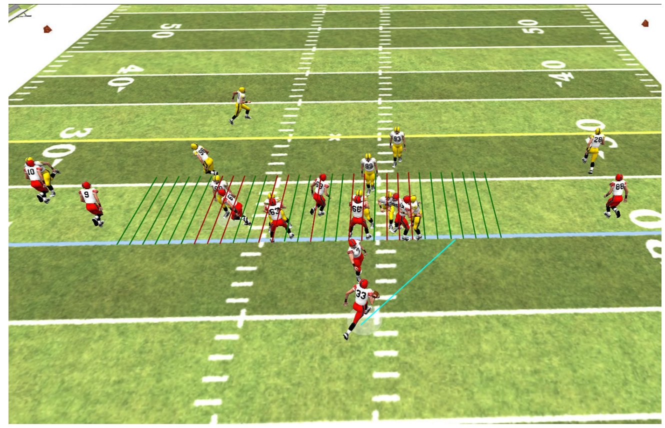

Figure 6: Increasing the distance at which ray casts extend towards the sideline will result in larger perceived holes on the outsides.

a high amount of diminishing returns in the accuracy of the raycasts when performed at intervals smaller than the approximate shoulder width of the players (four units).

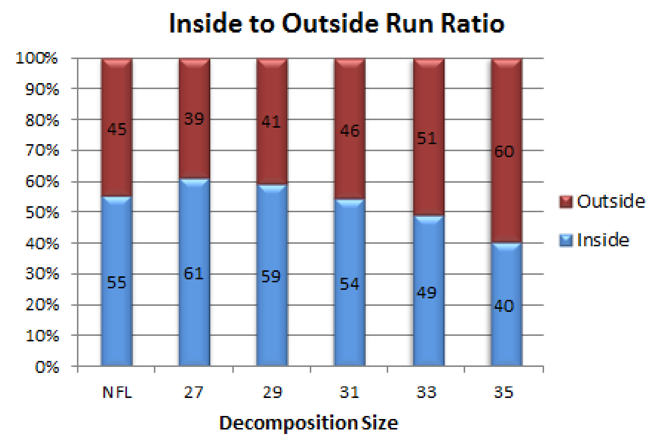

Figure 7: The result of 100 test plays at a variety of different decomposition sizes.

Once the appropriate raycast spacing was determined, the next goal was to specify the lateral bounds of the decomposition area — specifically regarding the horizontal distance that the raycasts were extended toward the sideline. Because part of the decision for selecting the hole is based upon its size, increasing the width of the decomposition area places extra emphasis on running outside of the tackle box (i.e., the area between the far left and right offensive tackles) since that area is not typically occupied by defenders at the time of a handoff and the larger perceived hole will be more attractive to the runner. That is not to say that one-sided holes will be ignored or under-emphasized. Indeed, figure 6 shows an environment state that presents an outside run as a desirable path option. Finally, because the intent of the algorithm is to produce behavior that closely resembles the decisions and actions of professional players, our goal was to closely match the number of inside (i.e., between the tackles) and outside paths that were ultimately taken with data compiled from the National Football League (NFL). According to Pro Football Focus, the ratio of inside to outside runs in the 2013-2014 NFL season was 55:45 [9]. Figure 7 shows the number of inside and outside runs that resulted from a variety of different decomposition sizes as compared to the NFL statistics. Each size was tested using 100 balanced rushing plays with an even mix of designed inside and outside runs. The player's decision to run inside or outside was recorded. We selected the 31 raycast decomposition size as it produced results that were the most similar to the NFL's ratio.

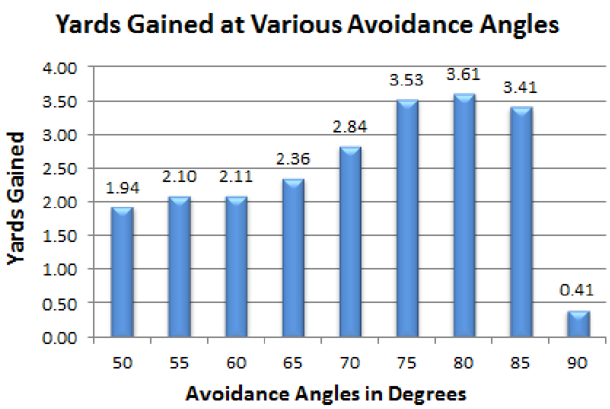

Figure 8: The average net yardage gain of 100 test plays run at a variety of avoidance angles.

Avoidance

The final set of tests executed was related to the AVOIDANCE substate. When the runner is attempting to avoid defenders, several tests were run to determine the angle — calculated by rotating the vector from the runner to the defender — at which to avoid the threat. Figure 8 shows the average net yardage (i.e., yards gained once avoidance is necessary) of 100 sample plays run at each avoidance angle from 50 to 90 degrees at five degree intervals. Angles at the lower end of the spectrum proved to be insufficient adjustments to threats, and a right angle avoidance approach not only yielded the lowest net yardage for the runner, but also produced illogical reactions when the runner faced a single defender directly in line with their forward vector. Our tests found the best approach was an 80 degree avoidance angle which yielded a higher net yardage gain for the runner than any other angle.

Conclusion

In this paper, we presented a framework for the decisions and actions of ball carriers in American football using the Axis Football Simulator. We divided the overall task of advancing downfield into two substates: HOLE SELECTION and AVOIDANCE, each with individual goals. During HOLE SELECTION, the runner attempts to find the best space between nearby players through which to run. Our tests conclude that a decomposition area of 31 raycasts, spaced at four units (roughly the width of a player) around the line of scrimmage, produces the most accurate results in identifying the location of players while minimizing the frequency of raycasts. Additionally, our tests proved the consistency of inside to outside runs that resulted from that decomposition methodology as it relates to National Football League data collected from the 2013-2014 season.

In the AVOIDANCE substate, the ball carrier attempts to avoid nearby threats by running at angles away from defenders. Our tests conclude that the best angle at which to avoid threats on a single side of the runner is 80 degrees. Overall, this framework provides specific and flexible methods for instructing ball carriers in games and simulations that utilize or implement the rules, regulations, and restrictions of American Football.

Future Work

For future work, in terms of providing a comprehensive framework for controlling all players in a football simulation, our work relating to ball carriers is only one piece of a much larger puzzle. While the Axis Football Simulation provides a set of instructions for all player positions that is consistent with actual football competitions, we believe that they can be iteratively improved to provide a more realistic simulation.

References

[1] Avelino Gonzalez and David Gross. Learning tactics from a sports game-based simulation. *International Journal in Computer Simulation*, 1995.

[2] Stephen Intille and Aa. Closed-world tracking. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 672–678, 1995.

[3] Stephen Intille and Aaron Bobick. A framework for recognizing multi-agent action from visual evidence. *American Association for Artificial Intelligence*, 1999.

[4] John Laird and Michael van Lent. Human-level ai's killer application: Interactive computer games. *AI Magazine*, 2001.

[5] Kennard Laviers and Gita Sukthankar. A monte carlo approach for football play generation. In *Proceedings of Artificial intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2010.

[6] Kennard Laviers, Gita Sukthankar, Matthew Molineaux, and David Aha. Improving offensive performance through opponent modeling. In *Proceedings of Artificial intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2009.

[7] Nan Li, David Stracuzzi, Gary Cleveland, Tolga Konik, Dan Shapiro, Matthew Molineaux, David Aha, and Kamal Ali. Constructing game agents from video of human behavior. In *Proceedings of Artificial intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2009.

[8] Allen Newell. *Unified Theories of Cognition*. Harvard University Press Cambridge, MA, 1990.

[9] ProFootballFocus. Staticstics from the 2013-2014 nfl season.

[10] David Stracuzzi, Alan Fern, Kamal Ali, Robin Hess, Jervis Pinto, Nan Li, Tolga Konik, and Dan Shaprio. An application of transfer to american football: From observation of raw video to control in a simulated environment. *AI Magazine*, 2011.