AdaptiviTree: Adaptive Tree Visualization for Tournament-Style Brackets

Desney S. Tan, Greg Smith, Bongshin Lee, and George G. Robertson

Abstract—Online pick'em games, such as the recent NCAA college basketball March Madness tournament, form a large and rapidly growing industry. In these games, players make predictions on a tournament bracket that defines which competitors play each other and how they proceed toward a single champion. Throughout the course of the tournament, players monitor the brackets to track progress and to compare predictions made by multiple players. This is often a complex sensemaking task. The classic bracket visualization was designed for use on paper and utilizes an incrementally additive system in which the winner of each match-up is rewritten in the next round as the tournament progresses. Unfortunately, this representation requires a significant amount of space and makes it relatively difficult to get a quick overview of the tournament state since competitors take arbitrary paths through the static bracket. In this paper, we present AdaptiviTree, a novel visualization that adaptively deforms the representation of the tree and uses its shape to convey outcome information. AdaptiviTree not only provides a more compact and understandable representation, but also allows overlays that display predictions as well as other statistics. We describe results from a lab study we conducted to explore the efficacy of AdaptiviTree, as well as from a deployment of the system in a recent real-world sports tournament.

Index Terms—Online fantasy sports, tournament, bracket, picks, adaptive tree visualization.

1 Introduction

Fantasy games, in which groups of people compete to predict outcomes in other competitions, is a large and rapidly growing industry. Jupiter research estimates that in 2005, consumers spent more than US$80 million on fantasy sports activities [9]. The Fantasy Sports Trade Association (FSTA) reported in 2006 [6] that there were 15 to 18 million fantasy sports players, with a growth rate of 7-10% each year. They estimate the annual economic impact of the fantasy sport industry to be US$1-2 billion. Another FSTA report estimates that each fantasy sport player spends about US$500 annually on magazines, online information, contests, and leagues [5]. These numbers do not even account for people playing fantasy games in other domains such as Academy Award or TV reality-show predictions. FSTA estimates that most participants spend about 3 hours per week managing their fantasy teams [6] and many online portals (e.g. ESPN, FoxSports, MSN, Yahoo!, etc) now run fantasy games in order to tap into this market of viewers.

Pick'em pools, which are also known as tournament or office pools, are a large component of the fantasy industry. In pick'em pools, players make picks, or predict outcomes, of real world contests in tournament-style competitions. Typically, players score fantasy points for each correct prediction and the winner is the person with the highest total score at the end of the tournament. Since players are unable to affect the real-world outcomes on which their fantasy scores depend, their participation is limited to observation, sensemaking, and scenario generation once the tournament begins. Players often look at tournament visualizations and try to make sense of how past outcomes have affected their standing and how future outcomes may change it. This allows them to choose which games to watch, which teams to support, and how to shape the social interaction that happens within the pool. Hence, the ease with which players can derive useful information from a tournament visualization is important and has direct impact on the amount of fun players have.

In this paper, we present AdaptiviTree, a novel tree visualization for tournament-style brackets (see Figure 1). AdaptiviTree deforms the representation of the tree and uses its shape to convey outcome information at each stage in the tournament. We assert that AdaptiviTree not only provides a more compact representation, but also conveys more information than classic tournament brackets. In the rest of this paper, we describe related work in tree visualizations, the current state of the art in tournament bracket design, the design and features of AdaptiviTree, lab study results showing the efficacy of deriving useful information with AdaptiviTree, and feedback from a deployment of the system for a recent sports tournament. We close with discussion on how the approach of adaptively changing the shape of a tree visualization to match and convey semantics can be generalized to problems in other domains.

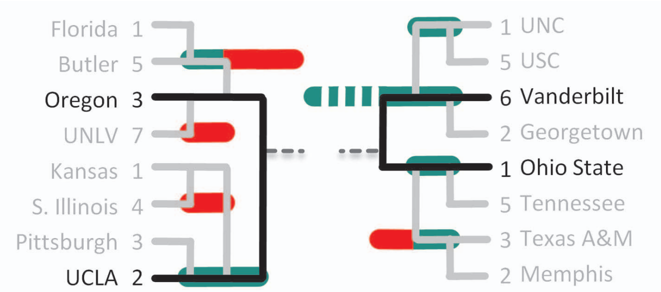

Figure 1: AdaptiviTree deforms its shape at each point in time in order to represent the data most effectively. It represents correctness of player picks with colored bars.

2 Background

In this section, we provide background on the terminology we use within the problem space, describe related work in tree visualizations, and discuss current state of the art in tournament visualizations along with the opportunities for improvement.

2.1 Terminology

A *tournament* refers to a number of *competitors* from a single sport (or other domain of competition) vying to be crowned the overall champion. Depending on the particular tournament, a *competitor* can refer to a single person (e.g. athlete), or a group of people (e.g. team). Each tournament consists of a sequence of head-to-head *contests* (sometimes referred to as matches, ties, fixtures, or heats)

• *Desney S. Tan, Greg Smith, Bongshin Lee, and George G. Robertson are with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-Mail: {desney, gregsmi, bongshin, ggr}@microsoft.com.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

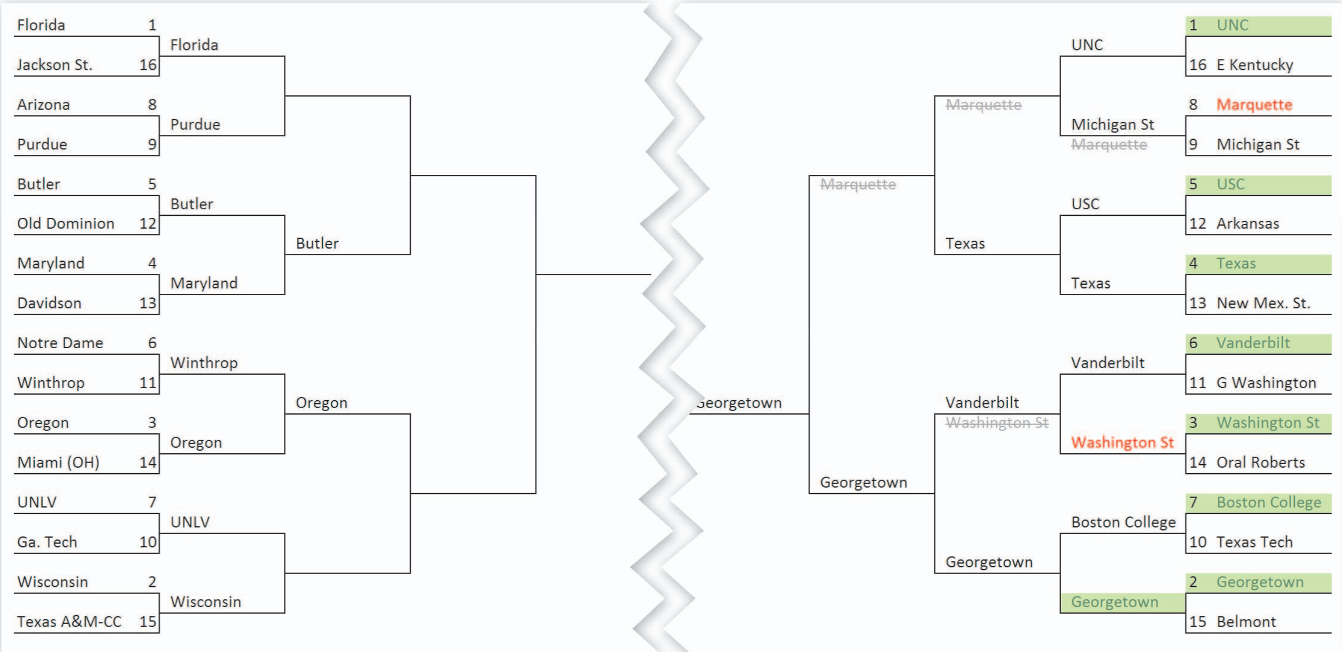

Figure 2: An example of a classic tournament bracket. We have broken the bracket in half, split vertically down the middle, in order to illustrate the way real-world contests are represented within the structure (left) but also the way that picks can be simultaneously displayed within the tree (right). In the right view, green boxes indicate correct picks, red text indicates incorrect picks, and grey text with strikethrough indicates picks that cannot possibly be right since the competitor has already been eliminated.

between competitors that lead to some result (i.e. one competitor winning and one losing). The basic goal of a tournament is to winnow multiple competitors down to a single champion, which often makes a tree, or hierarchy, a useful visual representation. In a single elimination tournament (also known as a knockout or sudden-death tournament), competitors who lose a match are immediately eliminated from the tournament (or at least from winning the tournament), and only winning competitors move on and vie to be the champion. For the purpose of simplifying discussion in this paper, we assume that we are describing single elimination tournaments when we say tournament. A *bracket* is the common term for a tree-based tournament visualization. The structure of the bracket defines how and when competitors will play each other as they progress through the tournament towards the championship.

We use the following terms to refer to elements of the fantasy games themselves. We use the term *player* to refer to a person who is taking part in the fantasy competition – in other words, a person who has completed a set of contest predictions. This should not be confused with a single-person tournament competitor, who is often also called a player in colloquial language. These tournament contest predictions are called *picks*. Most pick'em tournaments require that players pick all outcomes within the entire bracket. Multiple players who are competing against each other in the pick'em game form a *pool* or league. While pools can comprise just two players, or thousands, most typically include less than 50 players.

2.2 Tree Visualizations

Hierarchy, or tree, visualization techniques have been in use for almost three decades. Wetherell and Shannon [20] describe one of the earliest algorithms for tree layout on a display. Research on improved techniques for hierarchy visualization has been a key topic throughout this time, and has included a variety of 2-dimensional [1,10,16] as well as 3-dimensional approaches [13,18]. In particular, we were inspired by work on Phylogenetic trees (e.g. [15]), which demonstrate evolutionary progression and which share many properties with our problem domain.

The trees that are visualized sometimes get quite complex and researchers have explored adaptive layout techniques to help manage this complexity. These layout techniques such as Degree-Of-Interest Trees [1] and SpaceTrees [16] are often special cases of Furnas' Generalized Fisheye Views [7]. In these cases, nodes of interest are identified and a degree of interest (DOI) function indicates which nodes to retain in the layout, how large they should be, and which nodes to hide. The DOI function can pay attention to such nodes as the selected nodes, their parents, siblings, and parents' siblings. Another adaptive layout technique uses hyperbolic space [10,13] to control which nodes are displayed and how large they are. In this case, the selected node is centered and nodes far away are smaller or elided from the layout. In both of these approaches, the layout decisions are primarily syntactic. That is, they rely on the parent/child and sibling relationships rather than any semantic relationship between the nodes. In AdaptiviTree, we explore an adaptive layout that uses semantic relationships as well as syntactic relationships: winning competitors have the primary link going up the hierarchy, and losing competitors have a subordinate link. That is, the tree layout is adapted to the semantics of who has won or lost.

Some recent work on hierarchy visualization has focused on how to show changes in a hierarchy over time. One example is TimeTree [2], which adds a timeline to the DOITree approach, allowing the user to view changes to the tree over time. However, any particular view of the tree only shows data at a single point in time and the user must sequentially view time slices to see how the tree has changed. In contrast, we integrate changes over time into a single view in AdaptiviTree. Because the adaptive layout shows the winning competitor for each contest, the layout at any point in time represents the entire history of the tournament.

Other work has focused on comparisons of multiple trees. For example, TreeJuxtaposer automatically matches nodes in two trees based on the shared ancestors, and highlights where the differences are [14]. Other visualizations, such as TaxoNote [12] and CandidTree [11], merge the visualization of multiple trees into one. A subproblem of AdaptiviTree is to compare two trees, how the player picks correspond to real world contests, and our visualization for this builds upon much of the previous work.

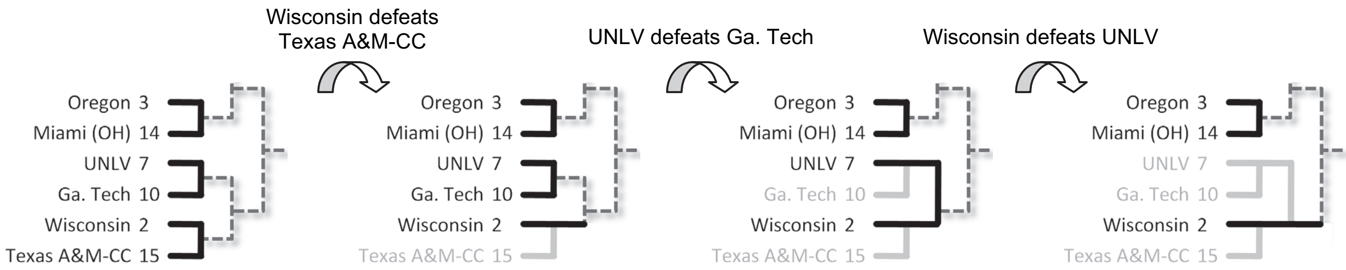

Figure 3: Time progression showing AdaptiviTree as more data is available. Note how the shape of the tree changes with each addition.

2.3 Visualizations for Sports

Many information visualization systems use sports data and statistics as motivating examples (e.g. [17]). For example, researchers have used parallel coordinates to visualize football statistics [19] and Breakdown Visualization to support analysis of sports statistics [3]. However, there has been little research specifically exploring how to effectively visualize sports statistics or results. One example, SportsVis [4], uses a bar display to visualize the individual team's performance by assigning game attributes to visual attributes such as color and length. It also uses a TreeMap to show the efficiency of playing time allocation for individual baseball players. Unfortunately, this visualization cannot be applied to tournament-style brackets. Previous work on sports competition visualization includes TennisViewer [8], which uses a TreeMap to show the results of all of the games in a tennis match. TennisViewer also integrates changes in the match over time into a single view. However, this system was not evaluated and it is difficult to determine if users of the visualization were able to effectively interpret it.

2.4 Classic Bracket Visualizations

Classic bracket visualizations utilize a static representation of the tournament structure throughout the course of the tournament. This structure contains a series of match-ups, which has competitors who are playing each other connected by a vertical line (see Figure 2). The winner of each contest match-up progresses to the next round and their name is filled in on the result line coming out of the match-up. This forms a new match-up with another competitor, and this process continues until there is only one competitor left, the tournament champion.

Most classic bracket visualizations offer a horizontal tree-view to make competitor text labels more readable. To achieve a reasonable overall aspect ratio, both for paper and electronic displays, many classic brackets divide the tournament into two sub-trees anchored at the championship node that mirror themselves around a vertical axis. In such brackets, competitors begin at the extreme edges of the bracket and progress towards the center, where the final contest is played and the champion determined (see Figure 2 left).

The classic bracket visualization was designed for use on static media such as paper. When using such media, it is critical that the bracket representation be incrementally updatable in a purely additive manner. As contests are decided and real world results are tallied, updating the bracket merely requires that winning competitors be inserted into the bracket.

This classic bracket view has been used for decades and is familiar even to people who do not follow sports and tournaments closely. Unfortunately, this visualization has at least two undesirable properties. First, it takes a significant amount of horizontal space since names are repeated for each round that a competitor progresses. We know of no straightforward solution that will compact this representation. Second, it is difficult to get a quick overview of the tournament state since the path of any given competitor goes through arbitrary up and down offsets with each round. This is inevitable since there is no way to predict the path of the competitors prior to acquiring the end results. Additionally, this winding path through the tournament requires reading repeated text as each competitor's name is repeated multiple times in previous rounds. The winding path also makes it difficult to design overlays that add information to the basic bracket. These overlays are important if the player would like to simultaneously see, for example, their picks or aggregate statistics for other players overlaid on the tournament bracket.

Since one of the primary tasks users try to do with tournament brackets is to make and track picks in relation to contest outcomes, most classic bracket views present tournament outcomes and picks on the same tree. Most of these visualizations leverage the fact that picks can be conveniently represented exactly like a completed bracket, with each prediction corresponding to an outcome. The role of the combined view is to present the differences between picks and the real world as contest outcomes become available.

There are various implementations of this, but most classic brackets share the following attributes. Correctly picked competitor names are turned green (and possibly bolded or boxed) to represent an exact correspondence between prediction and reality. These typically score the player points. An incorrect pick is turned red or grey (and possibly bolded or boxed), and often dual-coded with a strikethrough. The real-world winner of the contest is commonly added above or below the player's incorrect pick. This encoding adds an additional constraint to the minimum vertical space required in the classic bracket layout. If the player's incorrect pick was predicted as a winner of any future games, it is turned red (or grey) and given a strikethrough in those locations as well to reflect that even though the pick is for a future contest, the incorrect pick is now eliminated from the tournament and cannot possibly score the player any more points. For an example of one implementation of this, see Figure 2 right. Unfortunately, this encoding is somewhat complex and difficult to follow and compare, even for the most expert users.

3 AdaptiviTree Design

AdaptiviTree is designed for trees which change as new information becomes available. This makes it appropriate for visualizing tournament-style competitions, where the outcomes of the individual contests that make up the tournament arrive over time.

AdaptiviTree exploits the dynamic nature of computer-based visualization and does so in a way that allows much more information to be gracefully overlaid. The basic insight in AdaptiviTree is that the bracket tree structure can be deformed to present tournament outcomes in a non-textual way.

3.1 Basic AdaptiviTree Bracket

The initial view of the tournament is similar in AdaptiviTree as in classic brackets. However, rather than repeating competitor names on a static tournament bracket to indicate wins in each match-up, AdaptiviTree moves the result line up or down from a match-up to extend the incoming line of the winning competitor. Adapting the structure of the bracket rather than repeating the names to convey outcomes means that each competitor's name appears only once in AdaptiviTree (on the periphery of the bracket) and a competitor's progress through the tournament is represented by a single horizontal line extending from its name towards the championship. When a competitor loses, the horizontal line terminates, since the

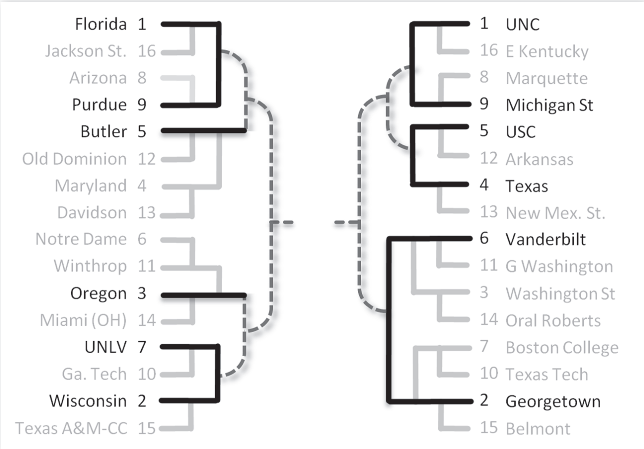

Figure 4: AdaptiviTree after 20 games have been played in the tournament. The shape of the tree is deformed so that relevant information is emphasized. Note how much smaller this representation is than the classic tournament bracket.

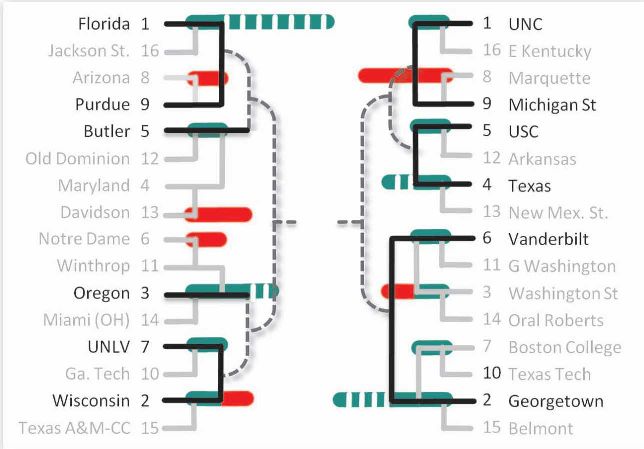

Figure 5: AdaptiviTree with picks and pick results overlaid on the bracket. Solid green bars represent correct picks, dotted green bars represent picks that could potentially be correct, and red bars represent picks that are confirmed incorrect.

continuation of the line in the next round moves to the winning competitor. In practice, we simply iterate through all results shifting lines up or down in order to attain the desired tree.

Since all the competitors are in (or alive) at the beginning of the tournament, we make the horizontal line for the first round shorter than others. This line represents the presence of the competitors in the tournament and not a win.

To make the visualization even clearer, we grey out lines and names of competitors that have lost, and represent unplayed match-ups as dotted lines. Making these dotted lines rounded rather than bracket shaped seemed more aesthetically pleasing and made them more distinct, but we have left them as square brackets as preliminary user feedback suggested that this made it look more like a classic bracket and was more comfortable to view. See Figure 3 for a sample progression of AdaptiviTree as a tournament unfolds.

In practice we animate these transitions in order to make it easier for players to see what has happened. In the case where multiple games have happened between viewings of the bracket, we permit players to select a point in time with a slider. Although all information is represented in a single static view of the current time slice, it is sometimes useful to scroll through time. Since the shape of the tree evolves as more information is added, interactively inspecting the incremental (and animated) visualization across time provides useful information that helps players even more easily make sense of historical events.

By looking at a static slice of AdaptiviTree, the user is able to quickly see the contests that have been decided (grey lines), the results of these contests (each competitor has gotten to the round at which its horizontal line ends), all competitors that have lost (grey competitor names), competitors that remain in the tournament (black team names), the pending match-ups about to be contested (competitors connected by black vertical lines), and future contests in the tournament (dotted lines). AdaptiviTree is a very compact representation and works well even with small amounts of space. See Figure 4 for a sample AdaptiviTree.

We have explored many alternative designs, including shrinking the names of the losing teams to de-emphasize them, removing them altogether rather than graying them out, removing vertical lines and leaving only the horizontal ones of past played games, and so on. Even with these variations, the core idea of AdaptiviTree, that the brackets deform and create horizontal line paths from the team names representing progress of individual teams, remains the same.

3.2 Information Overlays

Since information for each competitor is represented horizontally from each name, it is easy to overlay simple graphs conveying statistics for each competitor in an AdaptiviTree. Examples include statistics such as the likelihood of a team to reach various rounds in the tournament, or the percentage of players in an office pool that have picked a certain team to get to each of the rounds. Perhaps the most interesting overlay is one which shows a set of picks and the correctness of these picks overlaid with the real world results. As previously discussed, picks and correctness more generically represent inspecting one tree against another and this visualization can be expanded to view multiple trees simultaneously.

In AdaptiviTree, the outcomes are encoded in the tree structure, so the pick tree can be overlaid on the real-world outcome tree to produce differences in the line structures of the two trees without resorting to textual encodings. The lines defining the pick tree are expanded and made semi-transparent, and then shape- and color-encoded as an overlay according to the following rules: Each line segment where the pick matches the real world outcome is turned green, and each segment where the pick doesn't match (or is eliminated from consideration due to a prior loss) is turned red. A line segment corresponding to a viable pick in a future contest (that is, a pick that is not already eliminated) is made dotted green to reflect the possibility that the prediction could still turn out correct but is not yet determined (see Figure 5).

The AdaptiviTree visualization has several characteristics we believe to be advantageous over the classic bracket. The structure in AdaptiviTree is competitor-centric: Each competitor's results are presented in a straight line out from the competitor's name, and the length of the line presents the success of the competitor or of the pick in a salient way. The longest result lines are the competitors that have advanced the farthest; the longest pick lines are the competitors the player has predicted the greatest success for; the longest solid green lines are the player picks that were most accurate and advantageous in the pool; and the longest red lines are the player picks that were most costly. The dotted green segments represent all the player's opportunities for future scoring in the pool. This competitor-centricity also allows for overlays of other team-based information (e.g. probability of success in each round, or aggregate pick counts in each round) which we plan to explore in future work.

3.3 Interactivity

There are various ways a player can interact with the AdaptiviTree visualization. A player may click on various information links within the visualization to bring up information about the team, the statistics from a particular game (past, present, or future), or other relevant information content.

The player can also manipulate content within the visualization itself. For example, while it has been assumed that the information that is driving the visualization comes from a computer-based source, this may not always be the case. For example, in order to make picks, a player starts with an empty bracket containing only the competitor names and structure and predicts which competitors they think will win. There are several ways a player can do this. First, they could click on the competitor name to indicate that the particular competitor wins a game. Each click indicates a single win, and players can do this for each contest in the tournament. Since one competitor is eliminated per contest, this would require a number of clicks one less than the number of starting competitors.

An alternative method for making picks is to drag competitor names into the bracket. For example, rather than clicking on a competitor twice in order to indicate that it wins two contests, the player could just drag the competitor name past the second round to indicate that it is picked to win all contests through the round on which it has been dropped. The bracket can be updated only when the drop is made, or interactively as the user is moving the competitor through the various rounds to provide real-time feedback on the action that will be performed if the user drops it at that given moment. If the updating is done in real-time, the lines will move up and down as the user drags, and the drop target is the entire column (from top to bottom) that represents the particular stage in the bracket. This is possible because the tree imposes constraints on where a competitor can end up and there is no ambiguity so long as the stage in the bracket is well specified.

4 CONTROLLED LABORATORY STUDY

We conducted a controlled lab study to explore how well people could learn and use AdaptiviTree as compared to the more familiar and classic bracket views. Note that the main goal of this study was to evaluate the visual design, so interactivity was not included.

4.1 Participants

Sixteen (5 female) volunteers from the Greater Puget Sound region in Washington state participated in the study. We screened participants so that about half of them were familiar with tournament brackets and half were not. Participants considered "familiar" reported filling out more than 4 brackets in the last five years, and "unfamiliar" participants had not filled out any. We also screened participants for color blindness and required normal or corrected-to-normal eyesight. The average age of participants was 34.6 (33.1 for males, 38 for females), ranging from 18 to 56 years of age. The experiment took about two hours and participants were given a software gratuity for their time.

4.2 Tasks and Materials

We divided the study into three parts, each representative of typical tasks that players may perform when viewing their brackets. The first task was fundamental *bracket understanding*. This task tested understanding of the basic structure of the tournament as well as the real-world events represented within the bracket. For this task, we generated brackets representing a tournament at some stage in play. For all tasks, we generated brackets using a pseudo-random process, weighing higher ranked teams more favorably in each match, in generating both real-world results and picks. We did this to ensure that there would not be an inordinate number of upsets and that the tournament outcomes would seem realistic. We asked participants to identify (1a) the lowest ranking remaining in the tournament, (1b) which teams a particular team that we selected had beat, and (1c) which teams a particular team we selected had lost to. For the first question, we asked people to name the rank instead of the competitors so that there would be a unique answer. For the latter two questions, we highlighted the team in question with a red box on the first round of the bracket so that participants did not have to perform a tedious visual search to find this (see Figure 6).

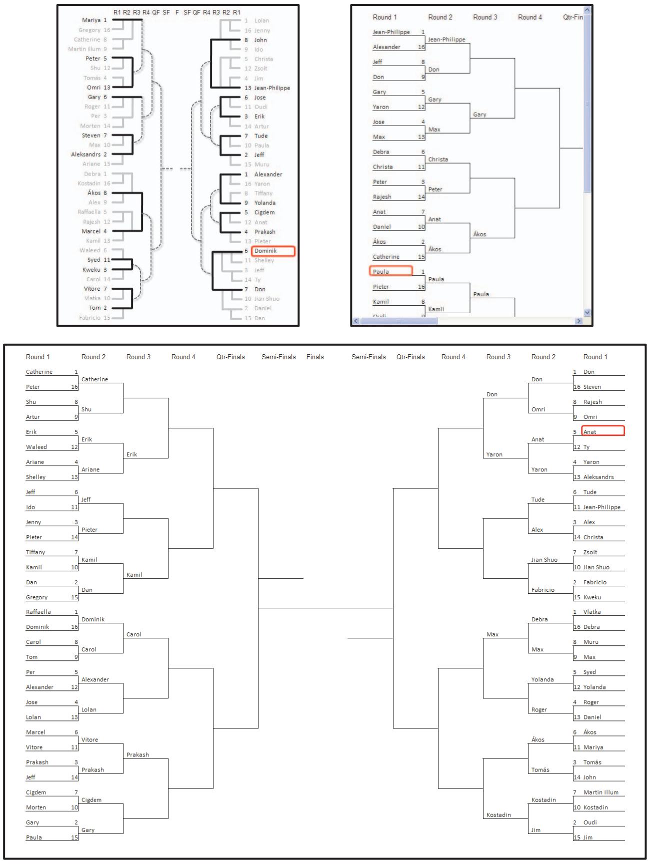

Figure 6: Three different Visualization conditions tested in the experiment – the AdaptiviTree (top left), space-constrained classic bracket (top right), and the classic bracket (bottom).

The second task was *pick understanding*. This task tested how well participants understood their picks and how well they corresponded to the real-world events. We generated brackets as well as a set of picks and presented this additional pick data within the bracket. In this task, we had participants identify: (2a) which teams had scored the player the most points so far (i.e. pick for this team corresponded to real-world results better than for any other team), (2b) which teams had the potential to score the most points in the future, (2c) which team was the biggest upset in the picks (i.e. the biggest difference between how far they picked the team to go and how early the team lost in the real-world).

The third task was *pick comparison*. These tasks tested how well participants could compare sets of player picks and derive useful information from looking at multiple brackets. In this task, we had participants simultaneously compare four brackets with picks and identify (3a) which of the brackets had the highest score, assuming each correct pick was worth a point, and (3b) which of the brackets had picked a specified team to go at least as far as player 1 had picked them to go. We repeated the latter question twice, with two different teams.

4.3 Visualizations

We compared three visualizations: AdaptiviTree, Space-constrained Classic Bracket, and Classic Bracket. We chose these conditions to represent the two optimal sizes for the AdaptiviTree and Classic Bracket visualizations, 480×600 pixels and 1280×1024 pixels, respectively. The Space-constrained Classic Bracket visualization simulates performance if a Classic Bracket were to be fit into the size of the AdaptiviTree. Since the Classic Bracket will not fit into a 480×600 pixel window, this visualization required the participant to scroll in order to see various parts of the bracket. To ensure similar readability, we used 10-point text across all visualizations.

For picks on the classic brackets, we used the Fox Fantasy MSN College Bracket (cbkgame.foxsports.msn.com) coding scheme. In this scheme, correct picks are turned green and incorrect picks are turned grey with a strikethrough. The incorrect pick was also put below the real-world winner of the contest.

We picked Animals and Cities as team names for these brackets and did not mention a particular sport or game because we did not want to evoke any preconceived notion of the tournament. The study was meant to compare the use of the visualizations and not rely on any prior knowledge.

Since we were not testing the interactivity, all three visualizations were generated as images through a semi-manual process. Figure 6 shows sample brackets in the three visualizations used in the study.

4.4 Study Design and Procedure

We ran the study as a within-subjects design, with each participant performing all the tasks using all the visualizations. We ran the three tasks in the same order across participants as this allowed us to teach the visualization components for each bracket incrementally: first the bracket alone, then the bracket and picks, and finally comparing multiple brackets with picks. Before the experiment, we explained brackets and the general structure of tournaments. Before beginning each part of the test, participants received instruction and performed a representative practice task in order to familiarize themselves both with the task and interface. All practice and test brackets were 64-team brackets. At the end of each session, participants filled out a satisfaction questionnaire.

We ran the bracket and pick understanding tasks as 3 (Visualization: AdaptiviTree, Space-constrained Classic Bracket, Classic Bracket) × 2 (Team: Animals vs. Cities) × 2 (Stage: 40 vs. 58 games played) × 3 (Question) designs. We balanced Visualization, Team, and Stage independently. The three questions were always presented in the same order.

We ran the pick comparison task as a 2 (Visualization: AdaptiviTree vs. Classic Bracket) × 2 (Team: Animals vs. Cities) × 3 (Question) design. Since we could not create a paper-based scrolling version, we dropped the Space-constrained Classic Bracket condition from this part of the study. Also, we only ran this task in one Stage with 40 games played as it became somewhat difficult and time consuming after 58 games were played, even for participants intimately familiar with tournaments and brackets. Participants compared two brackets instead of four in the practice tasks.

4.5 Equipment

We ran three participants performing tasks independently in each session of the study. Each participant worked on a 3.6 GHz Hewlett Packard xw4300 computer with a single 21" Samsung SyncMaster 214t LCD display running at a 1600×1200 pixel resolution. We presented the stimuli for the bracket and pick understanding tasks on the computer display, and showed pick comparison stimuli on paper because we could not always fit multiple brackets simultaneously on the display. This would have worked for AdaptiviTree but not the others. All tasks were driven and results logged by software running on the computer. For each trial, participants clicked on a button to indicate that they were ready to begin. A question along with four to six possible answers would appear. If the stimulus was shown on the display, it would appear under the question at that point too. For the paper-based pick comparison task, participants kept the paper brackets on the table in front of them and referred to them when needed. Participants looked at the bracket or set of brackets and answered questions as quickly and accurately as possible.

For all the tasks, the software collected task time and correctness for each question. We use these metrics to estimate the efficacy of each of the visualizations.

4.6 Results

We present the results from the study in four parts. First we explore performance on the bracket understanding task, then performance on the pick understanding task, followed by performance on the pick comparison task, and finally we investigate preference measures.

4.6.1 Bracket Understanding

We analyzed the task time data for bracket understanding with a 3 (Visualization) × 2 (Stage) × 3 (Question) repeated-measures analysis of variance (RM-ANOVA). We had initially included the Team manipulation, but found that removing this factor did not change results and makes it significantly easier to explain, so we will collapse this factor in all following analyses. Also, we found that participant familiarity with brackets did not have a differential effect across conditions, and we will not discuss this factor any further.

We found a significant main effect of Visualization ($F_{2,28}$=33.96, p<.001), with posthoc tests indicating that all interfaces were different from each other. Most interestingly, AdaptiviTree significantly outperformed both Classic Bracket interfaces (p<.001 and p=.002 for the Space-constrained and standard versions). We corrected all multiple tests in posthoc analyses using the Bonferroni technique. See left graph in Figure 7 for an illustration of the means. We also found significant main effects of Question ($F_{2,28}$=102.18, p<.001), with the first question "find the lowest ranking remaining in the tournament" taking a bulk of the time (p<.001). This is not unexpected as this question required participants to examine the entire bracket rather than just a portion in the other two questions.

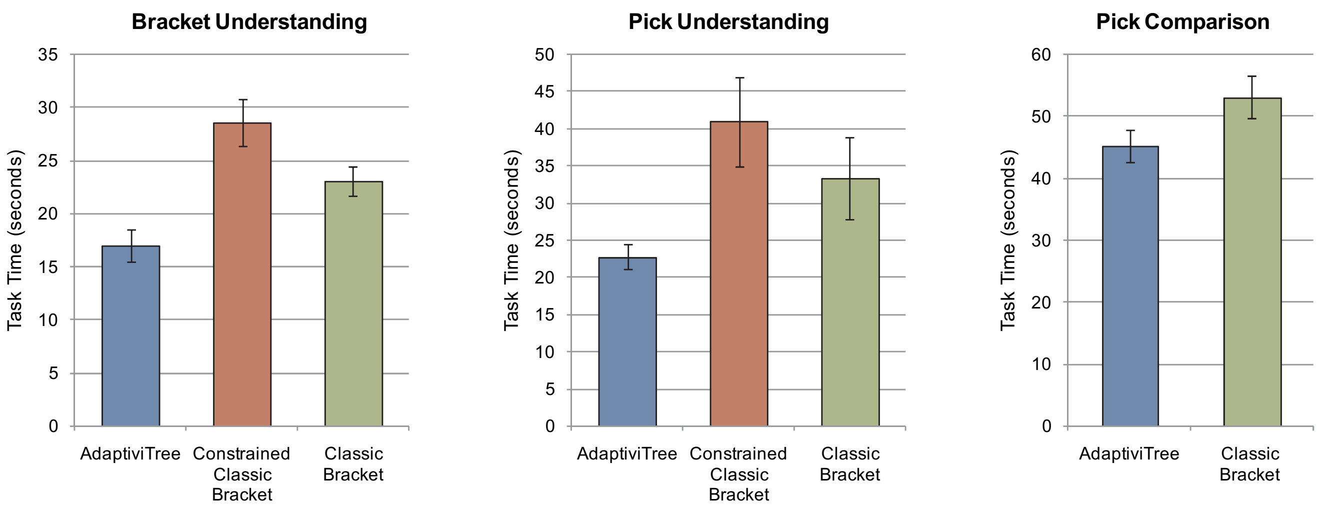

Figure 7: Task time means for the three tasks performed for each of the Visualization conditions. AdaptiviTree outperformed the other visualizations in all cases. Error bars represent standard error.

We also found an interaction between Visualization and Stage ($F_{2,14}$=6.11, p=.006), indicating that AdaptiviTree performed significantly better when moving from the brackets with 40 games played to 58 games played than either of the Classic Bracket visualizations. This indicates that AdaptiviTree scales to more complex brackets better than the Classic Brackets do.

We analyzed the Correctness of answers using a Friedman's Chi-Square test. Analyses showed similar patterns to the timing data. We found a main effect of Visualization ($\chi^2(2)$=15.94, p<.001), with AdaptiviTree leading to significantly fewer errors than the other Visualizations, and the Classic Bracket leading to fewer errors than the Space-constrained version. We found no other effects for the Correctness metric.

4.6.2 Pick Understanding

For pick understanding, we performed a similar 3 (Visualization) × 2 (Stage) × 3 (Question) RM-ANOVA for task time data. For this task, we found a significant main effect of Visualization ($F_{2,28}$=8.72, p=.001), with posthoc tests showing that this was driven by AdaptiviTree performing better than the Space-constrained Classic Bracket (p=.01). We also saw a main effect of Stage ($F_{1,14}$=8.99, p=.01), with the more complex 58 games-played bracket taking longer. There were no other effects for time. See center graph in Figure 7 for an illustration of the means.

For Correctness, we found a significant main effect of Visualization ($\chi^2(2)$=32.44, p<.001), with posthocs showing that the effect was driven by both AdaptiviTree and the Classic Bracket leading to fewer errors than the Space-constrained version (p=.001 and p=.004 respectively).

4.6.3 Pick Comparison

For pick comparison, we performed a 2 (Visualization) × 3 (Question) RM-ANOVA for task time data. Recall that since we did this on paper, the Space-constrained version of the Classic Bracket, which required scrolling was not relevant. Also, for this task, we dropped the Stage manipulation. We found a significant main effect of Visualization ($F_{1,14}$=5.84, p=.03), with AdaptiviTree performing better than the Classic Bracket. We also found a significant main effect of Question ($F_{2,28}$=21.78, p<.001), with the last question comparing the pick of a specific team across brackets significantly faster than the other two, which required more complex comparisons (p=.002). See right graph in Figure 7 for an illustration of the means. Correctness analyses revealed no differential effects across any of the manipulations.

4.6.4 Subjective Data

We ran Chi-Square tests for the subjective ratings and found significant results across most questions (p<.05). Participants indicated that AdaptiviTree made it easier to understand what has happened in the tournament, to detect how well (or poorly) picks were doing, to compare multiple brackets, liked the way it looked and enjoyed using it more than the Classic Brackets. They were also more favorable towards the full version of the Classic Bracket than the Space-constrained version which required scrolling. Curiously, when asked which of the brackets they would use in the next tournament they played, participants were divided, voting only marginally in favor of AdaptiviTree (9 votes to 7). We attribute this to familiarity with the Classic Brackets, which showed up strongly in the freeform comments.

5 Deployment Feedback

We also conducted a deployment of the AdaptiviTree visualization to get feedback of our system in relation to classic brackets. This deployment coincided with the 2007 NCAA Men's College Basketball tournament. Because we wanted to compare this to a commercial system and to leverage the setup and tracking capabilities of such a system, we tethered our system to the Fox Fantasy MSN Bracket Challenge (cbkgame.foxsports.msn.com). In collaboration with the Fox-MSN team, we fed the game and pick data into our system to generate a parallel website. The main difference was that we replaced the bracket view with a representation using AdaptiviTree.

We recruited volunteers from the pool of Microsoft employees who had either already filled out their brackets or were willing to fill one out on this system. We distributed a survey to solicit their feedback and comments halfway through the tournament. Of the sixty or so users, the survey received responses from twenty-five (4 female) volunteers. The average age of these respondents was 30.6 years of age. Respondents indicated filling out an average of about 4 brackets (ranging from 0 to greater than 10) in the last five years, and reported using a variety of online sites (e.g. CBS Sportsline, ESPN, Fox Fantasy MSN, Yahoo!, etc) as well as paper in previous experiences.

AdaptiviTree fared generally well against its classic counterpart in qualitative questions about the ease and efficacy of use, as well as the general aesthetic, and these results closely mimicked the survey results in the lab study.

When asked to list the best and worst features of both interfaces, respondents generally found that AdaptiviTree made it easy to track the current state of the tournament, which is not surprising since the visualization makes these games visually salient. Respondents also favored overlays representing the performance of picks in AdaptiviTree, and several claimed that this was an important feature in tracking their brackets. Multiple respondents commented on how this was a more glanceable visualization and allowed more effective gleaning of the key information. The largest positive comment for the classic bracket was familiarity with this representation, even if respondents had not filled out brackets previously. Also, since picks were made on this version of the bracket, we believe that respondents also had more familiarity with it even if they came in not having seen either visualization.

Respondents provided much useful feedback on how they thought we could improve AdaptiviTree. For example, several commented that the bracket did not look as polished as the Fox Fantasy MSN bracket. They also commented that past games were a little more difficult to interpret since they had to trace lines back to the team name, and that we should consider highlighting important past games such as major upsets. Additionally, this initial deployment did not have any visual aids that allowed the user to quickly determine the round in which a pick ended. We have since created a faint color gradient in the background to create distinct vertical bands. Others had comments about red and green being problematic and needing a second encoding, perhaps shape. We will address these comments in future versions of AdaptiviTree.

6 Discussion And Future Work

Results from the lab study support our hypothesis that the adaptive layout of AdaptiviTree is effective for understanding the bracket, understanding picks, and comparing picks. Both the lab study and the field deployment support our hypothesis that the AdaptiviTree layout would be more aesthetically pleasing.

There are several likely reasons for this. The AdaptiviTree layout takes substantially less space than the traditional approach, allowing the same information to be shown in less than half the display space. For a 64-team bracket, AdaptiviTree requires only about a quarter as much space as a classic bracket (see Figure 7). In addition, the deformed layout enabled direct overlay of relevant information, allowing picks and pick results to be effectively displayed in the same space. With these encouraging results validating the visual design, we plan to evaluate the interactivity of the system with more complex sensemaking tasks as part of our future work.

Another reason we believe AdaptiviTree provides more utility over classic brackets is that it integrates changes over time into a single view. By showing the winning competitor for each context, the total layout at any point in time shows all of the winning competitors. This is in contrast to earlier work on visualizing

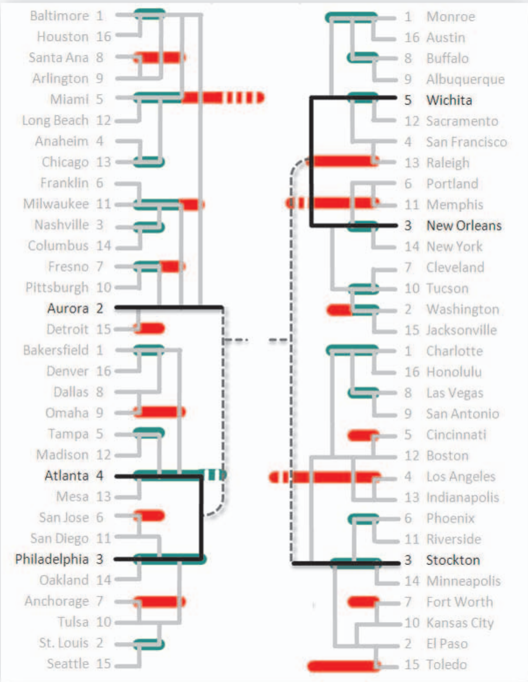

Figure 7: Example of a 64-team bracket with 58 games already played to illustrate complexity or clutter at this stage.

hierarchy changes over time with a timeline, which requires sequential viewing of the state of the tree at specific time periods. While the examples we have shown are binary trees and the distinguished node of each pair is the "winner," the AdaptiviTree technique can be generalized to any tree where one child of a parent is distinguished by some semantic property which is changing over time and which the designer wishes to emphasize. For example, this could be used to represent phylogenetics. We also believe that this technique can be extended to support more general cases where more than one child can be selected. Since the resulting structure would be a graph rather than a tree, we leave the specific design as future work.

This property of integrating changes over time into a single view depends on the syntax of the tree (e.g. the parent/child relationships) remaining constant. The AdaptiviTree technique could also work for arbitrary syntactic (parent/child) changes to a tree over time. However, the resulting layout would no longer show all of the changes over time, but would rather show only those parts of the tree that remained. For those remaining nodes, it would continue to enable direct overlay of relevant information.

7 Conclusion

In this paper, we have described AdaptiviTree, a novel tree visualization that adapts its shape to represent semantic information in tournament-style brackets. AdaptiviTree has several advantages over classic tournament brackets. First, it takes up much less space since competitor names do not get repeated in the tree, but rather are represented by the lines themselves. Second, because the semantics and information in AdaptiviTree lend well to glanceability, AdaptiviTree provides the player with a better overview of the data, and improves their ability to make sense of potentially complex relationships. We have described the design of AdaptiviTree and presented results from both a lab study and a field deployment which suggest that users are both more efficient and enjoy the task more when they are using AdaptiviTree. Finally we have discussed potential future work that could extend upon our current findings.

Acknowledgements

We thank Nirupama Chandrasekaran, Behrooz Chitsaz, Adam Eversole, and Gavin Jancke for their assistance building the deployment prototype; Tom Cohen, Jeff Johnson, Steve Lee, Mike Pendergraft, Krishna Veermachaneni, and the MSN team for providing access to the 2007 NCAA Men's College Basketball picks and game results; and Darko Kirovski for invaluable feedback.

References

[1] S. Card and D. Nation, "Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface," *Proc. Advanced Visual Interfaces,* 2002.

[2] S. Card, B. Suh, B.A. Pendleton, J. Heer, and J.W. Bodnar, "TimeTree: Exploring Time Changing Hierarchies," *Proc. Symposium on Visual Analytics Science and Technology*, pp. 3-10, 2006.

[3] N. Conklin, S. Prabhakar, and C. North, "Multiple Foci Drill-Down through Tuple and Attribute Aggregation Polyarchies in Tabular Data," *Proc. Symposium on Information Visualization*, pp. 131-134, 2002.

[4] A. Cox and J. Stasko, "SportsVis: Discovering Meaning in Sports Statistics Through Information Visualization," *Compendium of Symposium on Information Visualization*, pp. 114-115, 2006.

[5] Fantasy Sports Trade Association, "Fantasy Sports Conference to Unveil Demographic Survey and Discuss Legal Battle," *Press Release*, March 11, 2006.

[6] Fantasy Sports Trade Association, "Fantasy Sports Industry Demographics and 1st Annual Fan Choice Awards Announced," *Press Release,* March 24, 2006.

[7] G.W. Furnas, "Generalized Fisheye Views," *Proc. CHI Conference on Human Factors in Computing Systems*, pp. 16-23, 1986.

[8] L. Jin and D.G. Banks, "TennisViewer: A Browser for Competition Trees," *IEEE Computer Graphics and Applications*, vol. 17, no. 4, pp. 63-65, 1997.

[9] Jupiter Research, "Fantasy Sports: Programming for and Marketing to Hardcore Fans," *Report ENT05-C08*, December 29, 2005.

[10] J. Lamping and R. Rao, "Laying Out and Visualizing Large Trees using a Hyperbolic Space," *Proc. Symposium on User Interface Software and Technology,* pp. 13-14, 1994.

[11] B. Lee, G. Robertson, M. Czerwinski, and C.S. Parr, "CandidTree: Visualizing Structural Uncertainty in Similar Hierarchies," *Proc. IFIP Interact*, LNCS 4663, pp. 250-263, 2007.

[12] D.R. Morse, N. Ytow, D.M. Roberts, and A. Sato, "Comparison of Multiple Taxonomic Hierarchies Using TaxoNote," *Compendium of Symposium on Information Visualization*, pp. 126-127, 2003.

[13] T. Munzner, "H3: Laying out Large Directed Graphs in 3D Hyperbolic Space," *Proc. Symposium on Information Visualization*, pp. 2-10, 1997.

[14] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, "TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility," *ACM Trans. on Graphics,* vol. 22, no. 3, pp. 453-462, 2003.

[15] C.S. Parr, B. Lee, D. Campbell, and B.B. Bederson, "Visualizatios for Taxonomic and Phylogenetic Trees," *Bioinformatics*, vol. 20, no. 17, pp. 2997-3004, 2004.

[16] C. Plaisant, J. Grosjean, and B. Bederson, "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation," *Proc. Symposium on Information Visualization*, pp. 57-64, 2002.

[17] R. Rao and S.K. Card, "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information," *Proc. CHI Conference on Human Factors in Computing Systems*, pp. 318-322, 1994.

[18] G. Robertson, J. Mackinlay, and S. Card, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Proc. CHI Conference on Human Factors in Computing Systems*, pp. 189-194, 1991.

[19] J. Spacco, "Parallel Coordinates and Football: Multi-dimensional Data Visualization for Jocks," http://www.cs.umd.edu/class/spring2001/cmsc838b/Apps/presentations/Jaime_Spacco/index.html, 2001.

[20] C.S. Wetherell and A. Shannon, "Tidy Drawings of Trees," *IEEE Trans. on Software Engineering*, vol. SE-5, no. 5, pp. 514-520, 1979.