

Visualizing a Tennis Match

Liquan Jin and David C. Banks
Department of Computer Science,
Mississippi State University, Mississippi State, MS 39762
e-mail: {ljin, banks}@cs.msstate.edu

Abstract

This paper describes our work on visualizing the information of a tennis match. We use competition trees to organize the information of a tennis match and visualize the competition trees by the top-nesting layered maps with translucent colored layers. We create iconic representations to describe the detailed information of athletic events in an intuitive manner. Specialized views of the information are displayed by applying multiple Magic Lens filters on the top-nesting layered maps. The dynamic nature of the tennis match is depicted by the time-varying display. The approach we present in this paper can be used to visualize other sports information, information with competition property, or information with hierarchical structure.

1 Introduction

The graphic capabilities and speed of hardware systems have provided a strong base for exploring new techniques of information visualization in new areas. Sports data is one important part of our society but has not benefited from information visualization. Large data sets are generated from the athletic events day by day. ESPNET [3] (on the world wide web), radio, television, and newspapers are currently media that report sports events. They usually summarize a contest in top-down prose. They provide the general information (such as the final winner and the final score) at the beginning of

their report. After that, they describe the details of some interesting portions, then some sub-portions. They organize the sports information in a tree structure but display it in a linear manner. The match data and reports usually require many pages or screens to be displayed. Examining the sports information to extract the interesting parts or jumping from one item to another item far away is time-consuming and tedious.

There are commercial tools for displaying tree structures [6] and graphical summaries of football game (Figure 1) and basketball game (Figure 2).

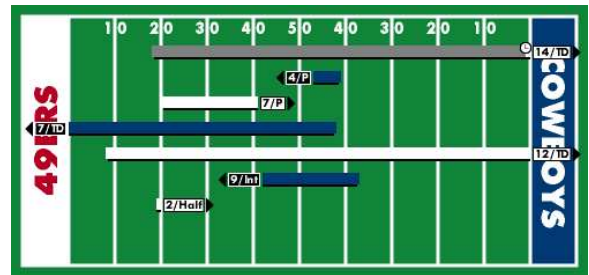


Figure 1: The summary of a football game.

In this paper, we describe our work on exploring visualization on sports information. We demonstrate our approach through visualizing tennis matches. We developed a tool, called TennisViewer, to provide an efficient interface for users to quickly explore tennis match information. Our main contributions are: (1) The top-nesting layer map supports visualizing the overall structure information as well as the detail information in one screen. (2) We represent athletic events to a detailed level. (3) We apply

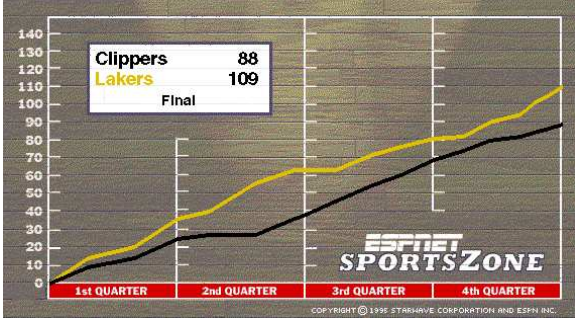


Figure 2: The summary of a basketball game.

multiple Magic Lens filters to explore specialized views of the information. (4) We provide a time-varying display to animate all or part of a match.

2 Competition trees

The information of a tennis match is hierarchically structured. One match consists of several sets. One set consists of several games. One game consists of several points. One point consists of one or two services. Each service consists of several strokes. Each level may contain different kinds of information. The basic information at the match, set, and game level is: (1) who wins and (2) what the score is. At the point and service level, a single bit of information (who win the service and the point) suffices. At the stroke level, the important information is the ball-trace: where the ball was hit, and where it was returned. Tennis matches have an important property: competition. Two players compete with each for each layer of the match hierarchy. It is meaningful to depict this competition property. The hierarchical structure, information, and the competition property form the basic elements of a tennis match to be visualized.

The hierarchically-structured information can be organized by trees. We can construct the trees from bottom (strokes) to top (the match), level by level. We think of the process of playing a match as a tree-building process. During the match, each player tries to build his own tree up

to the next level. When he completes a stroke, he builds a stroke node. If he completes the last successful stroke, he claims the node that lies one level higher: the service node. All his stroke nodes become the children of the service node. When he wins the (last) service, he claims a node that lies one level higher: the point node. The point node inherits all his service nodes as its children. When he wins the last point, he claims a node that lies one level higher: the game node. The game node inherits all his point nodes as its children. When he wins the last point, he claims a node that lies one level higher: the game node. The game node inherits all his point nodes as its children. This process continues, level by level, until the end of the match. At this time, the winner completes a tree by claiming the match node at the top level. Both players always compete for the next higher available node, but only one of them can claim it.

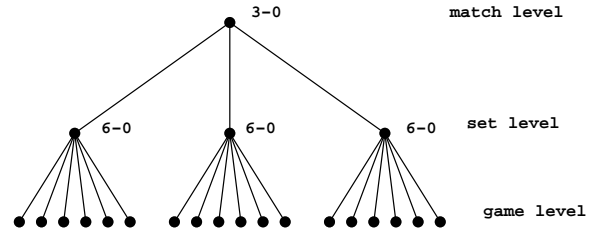


Figure 3: The top three levels of the ideal competition tree. Black wins every game, every set, and the entire match.

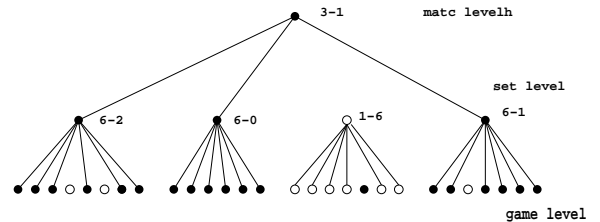


Figure 4: The top three levels of the priority competition tree. White wins one set and several games.

We combine the trees built by the two players together to be a competition tree. It depicts the hierarchical structure and the competition property of the tennis matches. Figure 3 shows the top three levels of the ideal tree for player

one (we use black dots to denote the tree nodes that he builds). He wins every game. Figure 4 shows the top three levels of the tree for a more typical match. We use white dots to denote the tree nodes that player two builds. Player one (black) wins the match but loses one set and some games.

We use the top-nesting layered maps and translucent colored planes to visualize the competition trees. The top-nesting layered map visualizes the hierarchical structure. The translucent colored layers visualize the competition property.

3 Top-nesting layered maps

Several techniques have been proposed for browsing complex sets of information. The Tree-Map method [5] offers a way to visualize hierarchically-structured information. It maps the full hierarchy onto a rectangular region in a space-filling manner. It is suited to depicting the structure of the hierarchy and the content of the leaf nodes. The Cone Tree [10] is another approach to visualizing hierarchically-structured information. It presents hierarchical structure in a 3D manner. Not all nodes are visible at the same time. Some nodes at the higher level may be hidden, but become visible by rotating the cone tree. The Perspective Wall method [7] is an approach to visualize information linearly. It uses hardware support to fold a wide 2D layout into an intuitive 3D visualization that has a center panel for detail and two perspective panels for context.

The top-nesting layered maps is organized using k-d trees [1]. It depicts the parent node and its child nodes by dividing the top-nesting rectangle associated to the parent node into sub-rectangles associated with the child nodes (figures 5 and 6). Each rectangle has a title bar to show information for the corresponding node. The space divides in a top-down process, from the match node to the stroke nodes. The match node holds the whole display space, the out-most rectangle. We vertically divide the rect-

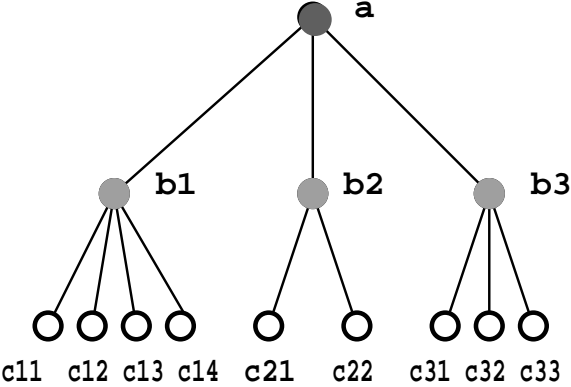


Figure 5: A tree

a		
b1	b2	b3
c11	c21	c31
c12		c32
c13	c22	c33
c14		

Figure 6: The top-nesting layered map of the tree in figure 5

angle under the title bar into sub-rectangles for sets. Then, we horizontally divide the rectangle under the title bar of each sub-rectangle for each game of the set. We continue the process level by level, alternating the dividing direction at each level, to build a top-nesting layered map. Figure 6 is the top-nesting layered map of the tree of figure 5.

The title bar of each rectangle provides a place to display information about the node. With an appropriate size for the title bar at each level, we can display a great deal of information compactly.

4 Translucent colored layers

The top-nesting layered map only shows the hierarchical structure and the short annotations. It does not depict the competition property. That is, it does not show who has won a node at a higher level. We assign colors to the rectangles to represent the competition. We use red to represent nodes claimed by one player and green for the other player. When a player builds his nodes in the competition tree, his color is assigned to the corresponding rectangles. Because the child rectangles are embedded in the parent rectangle, their translucent rectangles will reveal the color of the parent.

During the competition tree building, we put parent rectangles under child rectangles, layer by layer, for a total of six layers (stroke, service, point, game, set, and match). We make each intermediate layer translucent. Each intermediate layer is assigned an opacity value that ranges from 0 to 1. In order to reveal the lower level, we assign the opacity value in an ascending order. The color of each pixel is determined by blending the colors of the layers beneath it.

The upper image in figure 7 is the result after we blend the following game layer and the set layer into the match layer (bottom) in figure 7. We synthesized data for a tennis match between J. Bond (red color) and M. Michel (blue color). Figure 10 is the result of visualizing this tennis match. The match layer, set layer, and game layer of its competition tree are shown in figure 8. This top-nesting layered map depicts the hierarchical structure of this match. The scores of the match, each set, each game, the match title, and the players' names are shown in the corresponding title bars. The competition property is captured by colors. J. Bond wins the match with 3-2. The color of the match rectangle is red. He wins the first, third, and fifth set with 6-3, 7-6, and 6-2. The set rectangles of these three sets are shown as red-on-red. M. Michel wins the second and fourth set with 7-5 and 6-3. The set rectangles of these two set are shown as

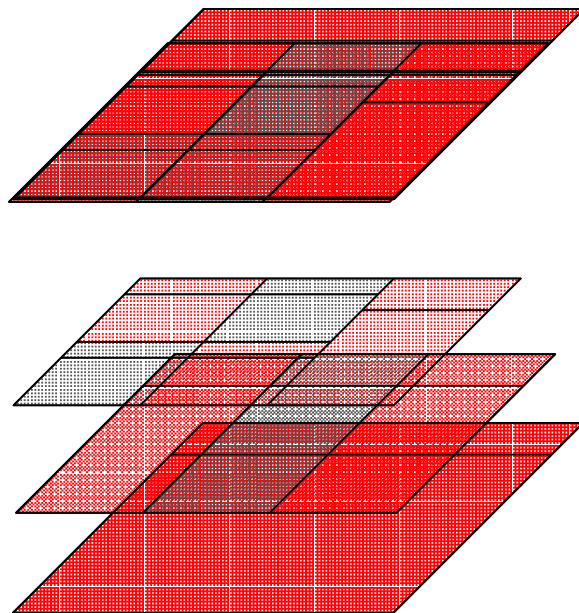


Figure 7: The top plane is constructed by projecting the following two planes into the bottom plane.

green-on-red. Because J. Bond wins the match, all the information in the display is colored by the global outcome of “winner = red”.

5 Iconic representation

In a tennis match, the ball traces of strokes are an important information to the coaches, the players, and the fans. Using a text description (Bond's service to the left court was returned out of bounds) is a laborious way to communicate the spatial information that describes the flight of a ball from one court to another. The ball traces are better to be depicted in pictures. We use an iconic representation at the stroke level to depict each ball trace. Each icon contains a tennis court; the red and the green colors represent the sides belong to the players, and the arrow shows the ball trace.

We place the icons in the middle of the TennisViewer stroke rectangles. The icons are resizable along with the display space allocated to the stroke rectangles. If the icon is too small or not

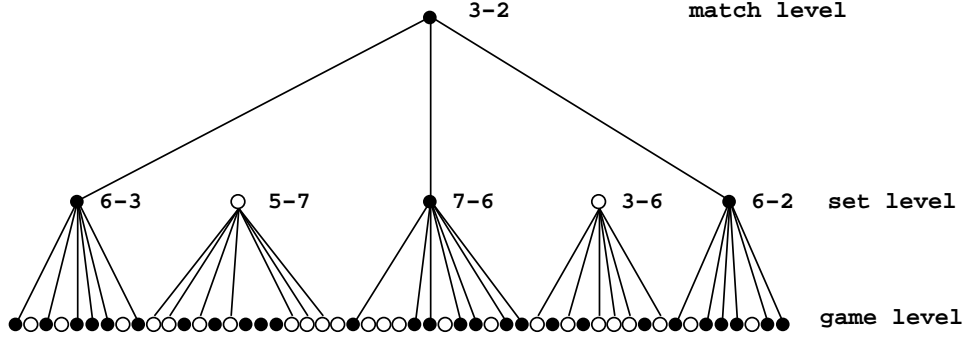


Figure 8: The top three layers of the competition tree of the Final of Golden Triangle Open (Black nodes are built by the red player. While nodes are built by the green player).

shown, the user can use the Magic Lens filters (discussed in the next section) to view them.

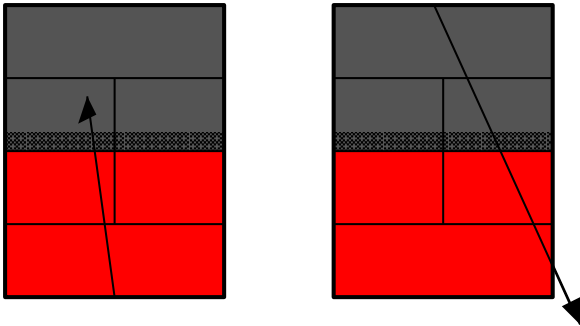


Figure 9: Iconic representation of ball traces. The return of the service goes out of bounds.

Figure 9 shows the iconic representation of the ball traces of a service. The ball is served from the bottom side to the top side. Then, it is returned from top side to the bottom side, but lands out of bounds.

6 Magic Lens

To reveal the information at the low layers of the hierarchy, we must magnify the image. A fish-eye view [4] is one commonly used method. It enlarges the interesting parts based on the distance of interesting. The motivation is to keep the overview of the information when enlarging the interesting part. The Table Lens [8] uses the

fish-eye method to browse large tables. The Document Lens [9] uses the fish-eye method to browse large documents.

A Magic Lens [2] opens a window over the image. It does not change the original image. It enlarges the interesting part in its window. Because the Magic Lens only draws the interesting part, it is fast and easy to update. By moving a Magic Lens on the image, the user can quickly explore the information. Moreover, Magic Lenses provide different filters to create special views of the information.

We adopted the Magic Lenses in our system to support exploring information at the low layers. The basic function of our Magic Lens filters is to zoom into the low layers. Through adjusting the zoom, the user can control the magnification of the Magic Lens filters. The user can open multiple Magic Lens filters at the same time when he wants to explore the information at different positions. The user can put one Magic Lens filter over other Magic Lens filter to enlarge the information in the bottom Magic Lens filter (figure 10). Manipulating the Magic Lenses is easier and faster than adjusting the zooming ratio because the adjusted zooming ratio may have to be re-adjusted later.

7 Time-varying display

The information of a tennis match is dynamic. At the beginning of the match, two players have no points and the competition tree is empty. As the match progresses, the players vie for points, games, and sets. They build their own trees from bottom to top. At the end of the match, one player wins the match and his competition tree grows to the match layer. Because some strokes, services, points, and games are more important than others, the display should reflect their importance as it dynamically updates to show information from a match in progress.

TennisViewer provides a time-varying display to support such animation. The user can select any part at any level to be animated. After the user selects the animated parts, our time-varying display rebuilds the part of the competition tree which is related to this animated part of the match and draws the top-nesting rectangles related to this part and colors these rectangles stroke by stroke from bottom to top, according to the rebuilt tree. When a player completes a stroke, the rectangle of this stroke is colored in the player's color. When the player wins a service, the rectangle of this service is colored in the player's color and is covered under the points of this service with translucent. When the player wins a point, this point's rectangle is colored in the player's color and is covered over the services of this point with translucent. The time-varying display shows when important events occur through color changes. The time-varying display also updates the title bars to enhance the information visualization. Figure 12 shows the effect when the red player wins the final point of the game of figure 11.

8 Exploring information

The tennis match information in figure 10 includes 1 match, 5 sets, 51 games, 332 points, 446 services, and over 2800 strokes. This is the first time that we have seen this much informa-

tion displayed to visualize a competition tree. The user can quickly locate the interesting sets, games, points, services, and strokes in this top-nesting layered map. He can capture the winners of each node by identifying red and green colors. He can locate the long rallies based on the density of the sub-rectangles. If he is interested in examining double faults, he can view the match through a Magic Lens that only displays double faults. The user can apply Magic Lens filters to enlarge the interesting parts to get a clearer view and use time-varying display to re-play the whole or part of the match. He can see each ball trace and the score changes.

We envision the use of video at the lowest levels to allow an actual replay of specific portions of the match. Future work includes installing statistical lenses (like spreadsheets) to operate on the data of the match.

9 Conclusion

Our system provides coaches, players, and fans, a new manner to analyze, review, and browse a tennis match. It captures a large amount of information in one screen. The user can quickly explore the information. Our approach is not limited to tennis matches. We select tennis because it is a popular sport, because there is so much information in a tennis match, and because it is organized in an obvious tree-structure. Our approach can be used to visualize other information with a hierarchical organization. For example, we can use our top-nesting layered map to visualize the directory structure of a file system for a collaborative project: player one owns a directory containing files which were created by player two. We can construct the top-nesting layered map from bottom files to the root directory. The names of the directories and the files can be displayed in the title bars. We can use icons or iconic labels to represent detail information.

References

- [1] J. L. Bentley and J. H. Friedman. Data structures for range searching. *ACM Computing Surveys*, 11(4):397–409, December 1979.
- [2] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of SIGGRAPH '93*, pages 73–80, 1993.
- [3] ESPNET.
<http://espnet.sportzone.com/tour>.
- [4] G. W. Furnas. Generalized fisheye views. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, April, 1986*, pages 16–23. New York: Association for Computing Machinery, Inc., 1986.
- [5] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization '91 Conference, San Diego, CA., October, 1991*, pages 189–194. Los Alamitos, CA: IEEE Computer Society Press, 1991.
- [6] Maryland Widget Library. *7200 Wisconsin Avenue, Suite 410, M/S 111, Bethesda, MD 20814-4811, USA, Email: mdwidgets@acm.org*.
- [7] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, April, 1991*, pages 173–179. New York: Association for Computing Machinery, Inc., 1991.
- [8] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, April, 1994*, pages 318–322. New York: Association for Computing Machinery Press, 1994.
- [9] G. G. Robertson and J. D. Mackinlay. The document lens. In *Proceedings of the ACM Symposium on User interface software and technology, November, 1993*, pages 1–4. New York: Association for Computing Machinery, Inc., 1993.
- [10] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, April, 1991*, pages 189–194. New York: Association for Computing Machinery, Inc., 1991.

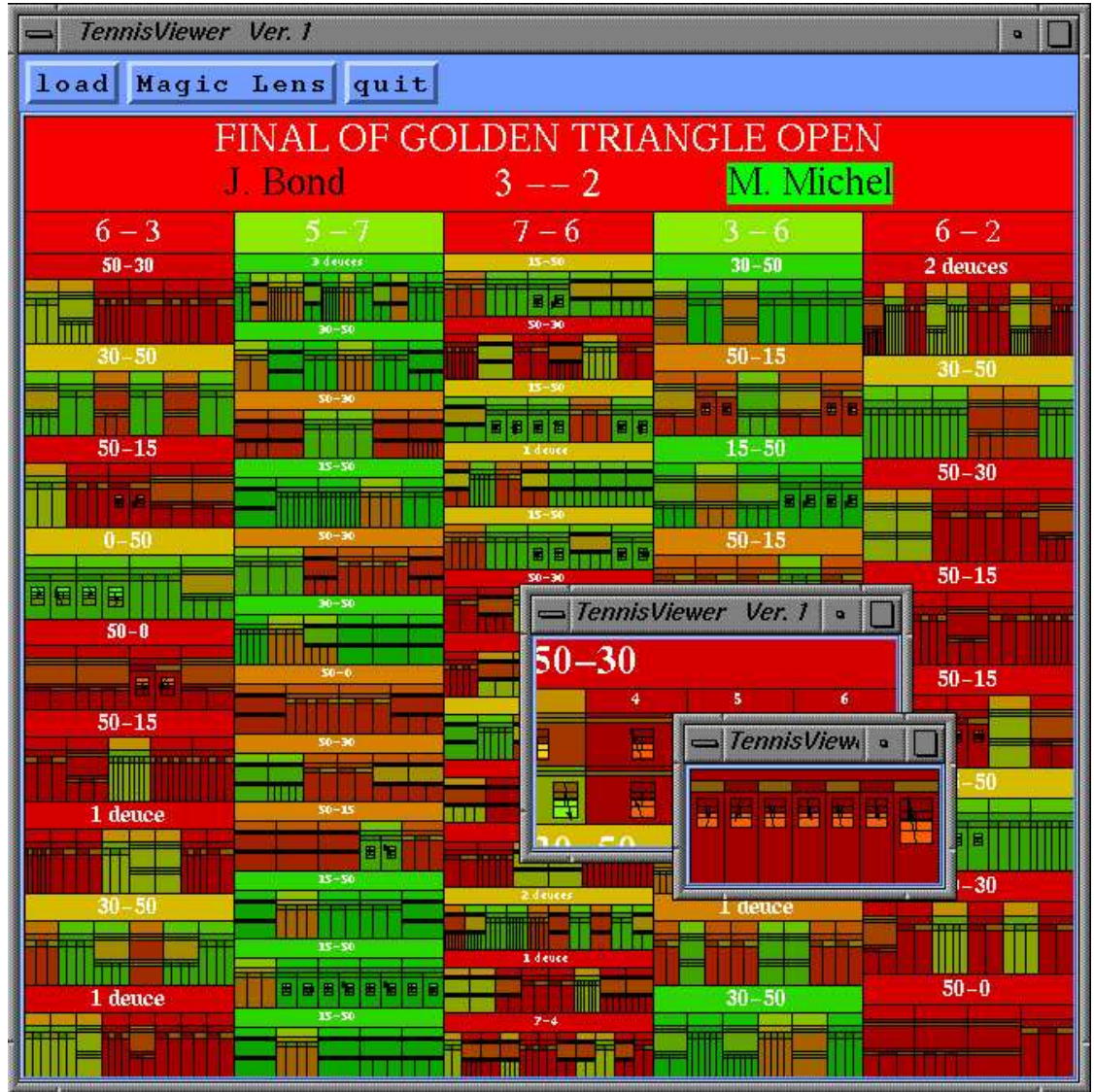


Figure 10: The top-nesting layered map of the Final of Golden Triangle Open. One Magic Lens filter is on the top of another.

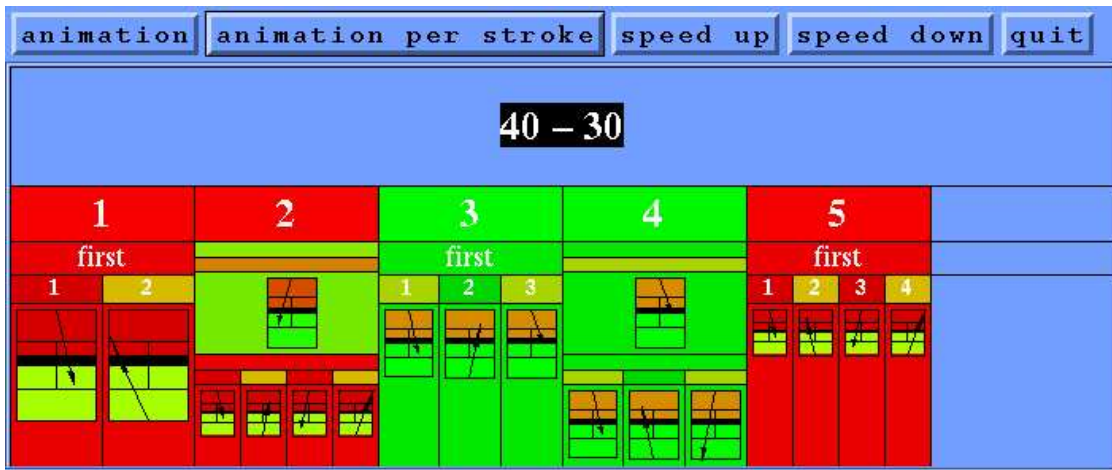


Figure 11: Animation – before the final point

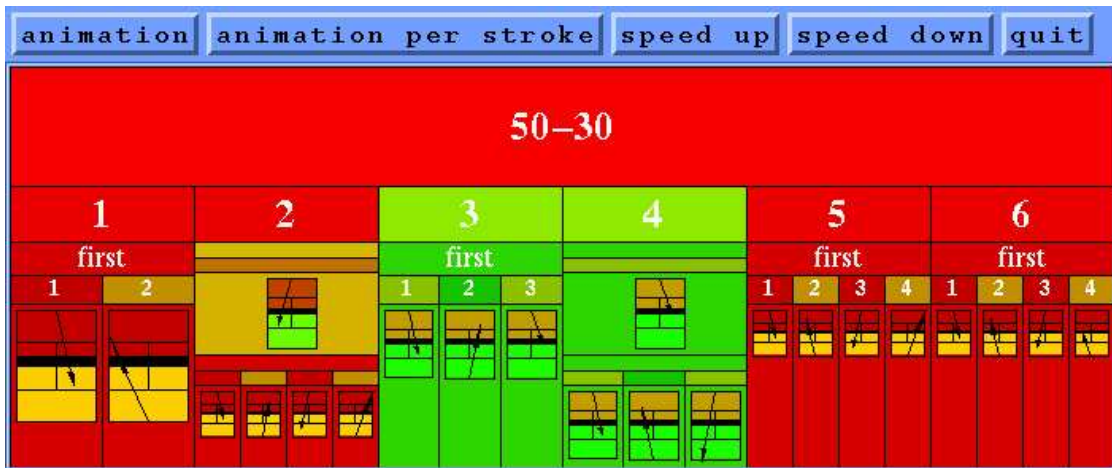


Figure 12: Animation – after the final point