Delft University of Technology
Data Visualization 2015 - 2016

Football Match Visualization Project
Individual Report

Overvoorde Alexander 4153235
January 14, 2016

1 3D visualization

I wrote a Python script that took the position, energy and speed data and generated team and per-player heatmaps from these as 2D grids with values scaled from 0 to 1. These are stored as arrays in JSON format. I visualized these with three.js (figure 1) by creating a plane and deforming it based on the heatmap value. The color is calculated using a d3.js scale. I created the field models (pitch, goals) with Blender. I implemented the camera movement by taking the code from the three.js OrbitControls example.

The players are simply cube geometry and are positioned based on the interpolated positions in the 2D visualization. They are also rotated based on the direction of the player. To make this work properly with the transitions I implemented a custom `attrTween` function that finds the shortest angle so that a player rotating from 350 degrees to 5 degrees doesn't turn all the way around.

Player selection is implemented by assigning the ID to each player mesh as JavaScript object property and using a raycast to find which player is under the cursor when the user clicks.

2 2D visualization

I developed the 2D visualization (figure 2) with d3.js and wrote the code for the playback controls and heatmap selection. The heatmap selection toggles the visibility of the generated heatmap meshes in the 3D visualization. The play/pause button and playback speed selector modifies the frequency of a `setInterval` timer that loads the next frame of player positions. The updates use d3.js transitions to make movement appear smoothly.

The frame data originates from a CSV file that is loaded with an AJAX request. The CSV contains the position, energy and speed per player per second. I generated it by importing the source data in PostgreSQL and exporting only the per-second data with the following query:

```sql
SELECT DISTINCT ON (DATE_TRUNC('sec', timestamp), tag_id) *, DATE_TRUNC('sec', timestamp) AS t
    FROM positions;
```

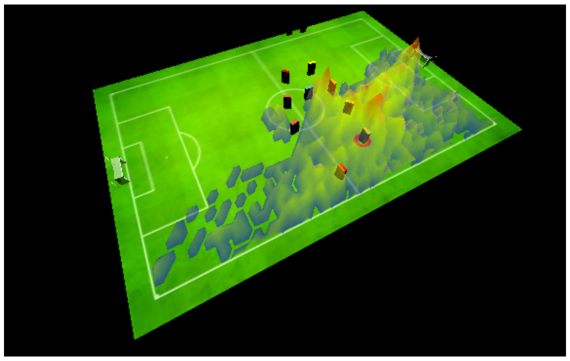
Figure 1: 3D visualization with heatmap overlay

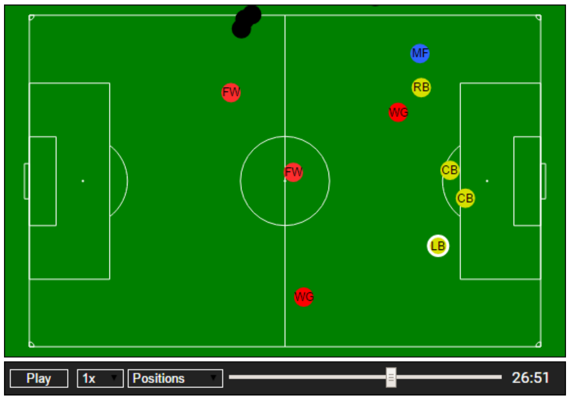

Figure 2: 2D visualization with playback controls