Multi-Touch Computing Mid-Project Report

Multi-touch Interactive Data Cube Visualization

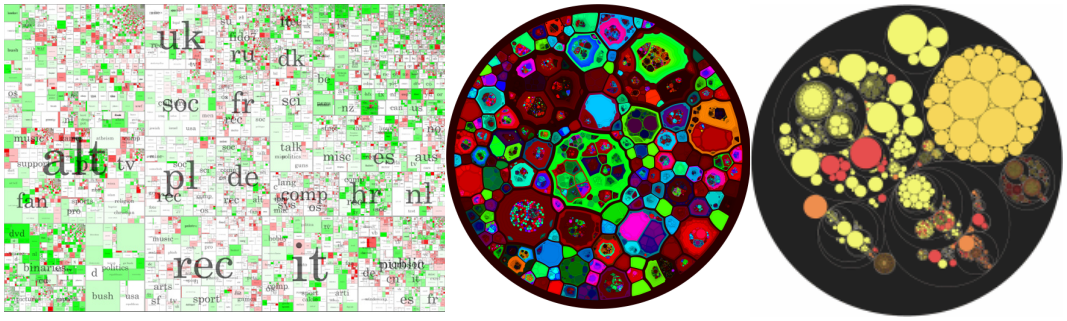
Curran Kelleher

4/14/2011

I proposed to develop an interactive multi-touch data visualization tool for interactively visualizing and navigating hierarchical data cubes. This overall concept has not changed, however one major departure from the original plans is to do the project on the iPad rather than the Microsoft Surface. This change has posed new challenges, but the project is progressing well nonetheless. I have worked on building up a demo on the iPad in which data is dynamically retrieved and visualized using the nested circles layout scheme originally proposed. The implementation of several more key features will lead to completion of the project by May 5.

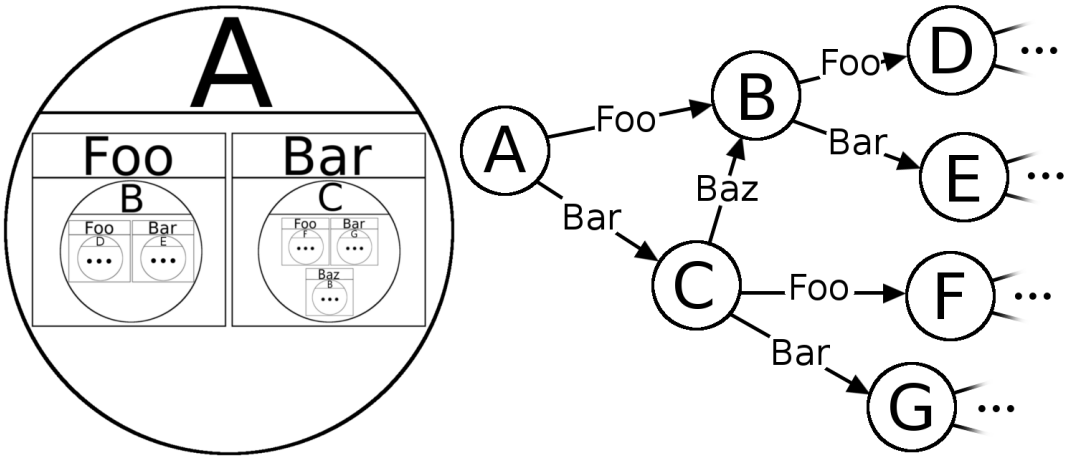
Developing for the iPad is completely different from developing for the Surface. I chose to use HTML5 rather than to develop a native iPad app in objective C, as being familiar with HTML5 and associated technologies will be advantageous for me going forward, as many are stating that HTML5 is the future of web based visualization. I used a library and language I am familiar with, Processing.js, which is an implementation of both the Processing language and immediate mode graphics API. This means I can code in a language that looks and feels like Java, and have the resulting programs run in web pages. Connecting the multi-touch events from the iPad Javascript API with the Processing graphics code was trivial, and once that was in place I was off and running.

Before embarking on implementation, I conducted a brief literature review to see if anyone has used nested circles to represent and navigate hierarchical data sets. TreeMaps are a visualization technique for hierarchies using nested colored squares. Voronoi TreeMaps are a variant which uses voronoi tesselationsrather than squares to represent hierarchies. Finally I discovered Circular TreeMaps, which are exactly what I am proposing. Now I know my work can be characterized as an application of circular TreeMaps.



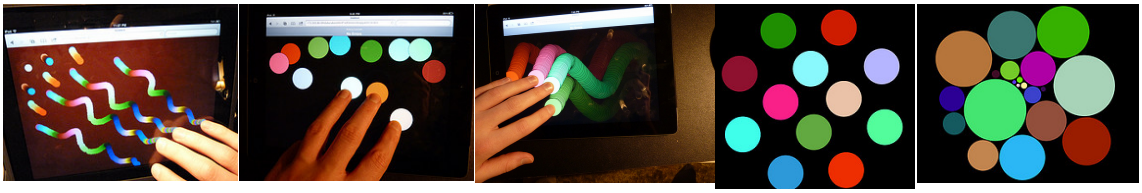
TreeMap, Voronoi TreeMap and Circular TreeMap.

Since the project proposal I had the idea to represent a view into a semantic graph using nested objects supporting continuous visual semantic graph browsing. This representation is a nested object visualization representing a view into the graph from a node's viewpoint. In this view, panning and zooming interactions drive graph traversal. I call the visualization technique Fractal Perspective, because it has approximate fractal structure (self similarity at multiple scales) and represents a kind of "perspective projection" from graph space to display space in which distant objects (in graph space) appear smaller (in display space). This is just an idea, but I hope to implement it in the future.



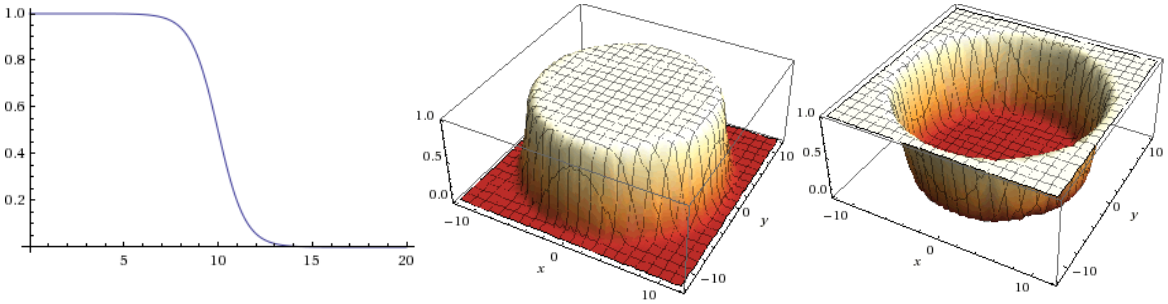
A mock up of the Fractal Perspective technique (left) and the corresponding node-link representation (right).

I built up a series of demos, posted at www.datacubes.info, which introduce progressively more capabilities and lead eventually to an interactive visualization. I coded in Processing from the ground up the following capabilities: tracking distinct touches using color, manipulating persistent circles, a force directed layout of circles (including inertia and bouncing off walls), and finally an interactive self organizing circle packing algorithm.



The series of HTML5 iPad demos chronicling development of the project to date.

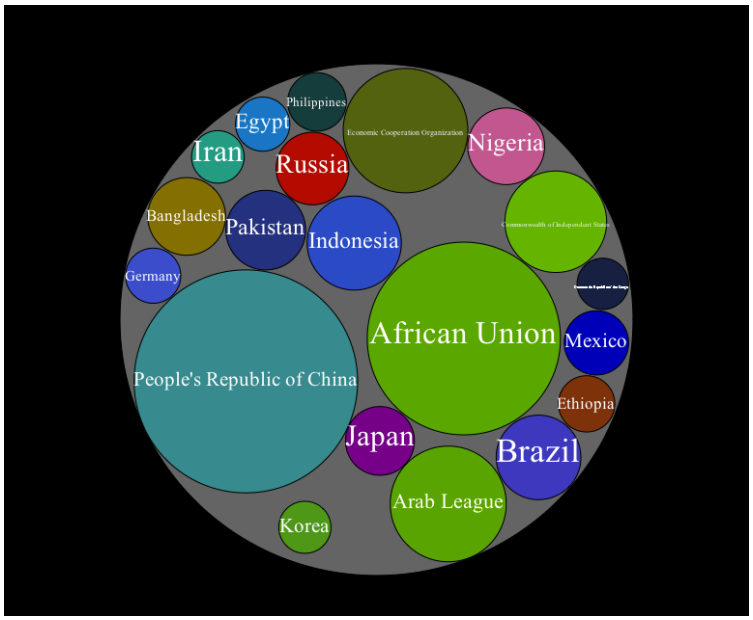
Many force directed schemes were explored for optimally laying out the circles, and the most successful one was based on the sigmoid function. If a threshold radius was used, the spacing would be correct but the system would continually undulate, as circles would oscillate between being pushed and pulled. A gravity function (c/r^2) converged smoothly but didn't result in correct spacing between circles (some would overlap and some would be too far apart. The sigmoid function, which defines a smooth transition from no force to substantial force, resulted in smooth convergence with proper spacing, as each circle pair could find that sweet spot balancing push and pull afforded by the sigmoid function.



The Sigmoid function (left), and its variants for repelling the inner circles (center) and packing the inner circles using the outer circle (right).

I learned how to query semantic graphs published on the web via a SPARQL endpoint. Resource Description Framework (RDF) is a general data model for representing semantic graphs developed by Tim Berners Lee and the World Wide Web consortium. SPARQL is the query language for RDF. DBPedia

is a project which scrapes Wikipedia content and generates from it a queryable semantic graph. In Javascript, one can easily query DBPedia and process the results.



The current state of the project: visualizing results from a SPARQL query against DBPedia. Circles are countries, and their area corresponds to their population. Color is arbitrarily assigned.

The final phase of my implementation to date was to query DBPedia for the top 20 most populous countries and visualize the result using the nested circles layout algorithm. This phase required figuring out how to draw labels properly on the circles and tweaking the layout algorithm a bit for optimal circle packing. Future development plans involve implementing panning and zooming, implementing multiple views, querying the Bureau of Labor Statistics data set, and implementing dynamic retrieval of successively deeper nesting levels. I believe I will be able to complete the originally proposed project by May 5.