3D Visualization of Hierarchical Clustered Web Search Results

Mehrnaz Sadat Akhavi [1], Mohammad Rahmati [1], Nerssi Nasiri Amini [2]
[1] *Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran*
[2] *Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran*
{Akhavi, Rahmati}@aut.ac.ir, Nasiri_amini@comp.iust.ac.ir

Abstract

The immaturity of semantic search engines has initiated researchers to apply various novel post-processing techniques on traditional search engine results among which clustering routines are most conspicuous. While many of these routines are focused on hierarchical clustering, little has been done toward an effective visualization of such data. Due to the richness of information observed in 3D in comparison with 2D (because of its abundant visual cues), we have proposed an intuitive 3D metaphor for the visualization of hierarchical clustered results. Our metaphor is based on fractal trees which are usually constructed by linear recursive algorithms traversing node-link hierarchies. The search results of different hierarchical clustering algorithms shall be visualized as either single-tree or forest metaphors which are user's customizable alternatives in our implemented prototype. In the forest depiction, our approach is even applicable to flat clustered results.

Keywords: *Search results visualization, tree visualization, hierarchical clustering, virtual reality, human-computer interfaces.*

1. Introduction

The information overload and the rapid growth of resources on the World Wide Web, makes search engines one of the most important and frequent services used to find required information over the net. However searching the web is sometimes frustrating, not only because of the underlying information retrieval technology but also because of the contemporary web-based user-interfaces. The most sate-of-the-art user-interfaces for the conventional search engines typically return long lists of ordered documents, ranked and displayed linearly in html format (usually 10 results per each page). Users are forced to sift through a great number of results which are not often relevant to their interests. A recent study [1] shows that 88% of users will try a new search if they do not find what they seek in the first three pages. Reviewing 30 results ordered by ranking mechanism which often does not reflect the users' preferences, makes them revise their search query or shift to another search engine. Therefore applying post-retrieval document visualization techniques seems to be helpful. Clustering the search results and improving the current textual user-interfaces into graphical are the main approaches in this regard.

Search results clustering algorithms attempt to group results together based on their similarities; thus results relating to a certain topic will hopefully be placed in distinct clusters. Clustering algorithms are distinguished by their implementation as flat clustering (only one-level partitioning of the data) or hierarchical clustering. Hierarchical approaches result in a tree-like construction where the clusters of closely related documents are nested within bigger clusters containing documents that are less similar. Organizing the huge amount of results to nested clusters, allows users to refine the search, starting from very general topics and moving towards the more detailed ones. Hierarchical clustering is so popular that it has been extensively employed in most industrial systems (e.g. Vivisimo, Mooter, IBoogie and Clusty), open source solutions (e.g. Carrot2) and research prototypes (e.g. Credo or SnakeT) [2].

Presentation interface is another important post-processing approach toward representing the search results in a more effective and intuitive manner. While a great deal of effort has been made to enhance the underlying post-processing methods, the visualization aspect has been less taken into consideration. Traditional browsers which allow the user to follow links in hypertext are unable to present the whole result in the available space. A graphical user interface (2D or 3D) helps users to see more results as a global view and to discern their relation. Unlike flat 2D representations, 3D visualization helps to present more information in the enlarged space by adding it an extra

Computer Graphics, Imaging and Visualisation (CGIV 2007)
0-7695-2928-3/07 $25.00 © 2007 IEEE

dimension. This enables us to exploit cognitive and spatial metaphors to illustrate large amounts of data in just one scene. Exploiting new interaction techniques and cognitive capabilities (such as changing the view point to improve perception) shall be considered as the advantages of 3D visualization.

Popularity of hierarchical clustering search engines and advantages of 3D information visualization, motivate us to propose an intuitive metaphor for visualization of hierarchical clustered search results in 3D space. To the best of our knowledge, there is no previous work reported in regard of this combination. *Tree* is a simple concept which is very familiar to everyone. It is a standard term for hierarchical information structure too. We propose a 3D tree metaphor as an intuitive and meaningful visualization for representing the structured search results. An advantage of our prototype is that the end user merely needs to use an X3D-enabled browser (a light plug-in for traditional web browsers) to access 3D scene. We have the use of mixed interface (combination of 3D scene and 2D interface) which is more successful than pure 3D worlds for search systems. Our metaphor is even applicable to flat clustered results.

The remainder of the paper is organized as follows: In section 2, a short overview of related projects is presented. In section 3, our prototype, Carrot2 clustering engine and our proposed metaphor are described. Section 4 presents the prototype application. Finally, section 5 summarizes the paper and gives an outlook on our future work.

2. Related works

In an investigation over document retrieval systems, Zamir has proposed a taxonomy of post-retrieval document visualization techniques [3]. According to this taxonomy, search results visualization techniques are divided into two main categories: The visualization of document attributes and the visualization of inter-document similarities. The first category, aims to display additional information (attributes) about the retrieved documents. VR-VIBE system [4] is an example of this type which visualizes documents in a 3D virtual reality environment. The main drawback of this system is small number of presented search results properties. Our approach does not deal with this category.

Visualization of inter-document similarities represents the links which can be followed to access the next document by the end-user. Graphs and maps are two main techniques to visualize such structures. In the graph approach, documents (web pages) are considered as nodes while arcs demonstrate the links between those nodes. Kartoo [5] is an example of a meta-search engine with a remarkable graphical user-interface which shows graph based documents relations in 2D space. The main drawback of this system is lack of results overview in the scene. In the map approach, users can take advantage of cognitive aspect. WEBSOM project [6] is an example of 2D landscape visualization based on map. The main drawback of 2D visualization of graphs and maps is that they lose their readability by increasing the number of results.

A recent work which is the closest approach to the one presented in this paper is SmartWeb [7]. SmartWeb project organizes the results using Kohonen self organizing map and visualizes them in a 3D space in the form of a city metaphor.

Another recent prominent work in 3D visualization of search results is Periscope system which applied the AVE method [8]. This system is based on adaptive visualization of the search results returned by indexing search engines.

Our approach deals with *clustering* which is a significant method applied in inter-document similarities visualization. The Galaxies visualization display in the SPIRE system [9] visualize flat clusters as a galaxy metaphor in which “suns” depict cluster centroids and dots scattered and interpolated near them mark individual documents. To the best of our knowledge, visualization of hierarchical clustered results (especially in 3D space), has been remained untouched.

3. Our prototype

In this section, Carrot2 clustering engine, simplified overview of our prototype conceptual schema and the proposed visualization metaphor are described.

3.1. Carrot2 clustering engine

Carrot2 [10] is an open source framework for research experiments, especially in clustering search results domain. This system is a post-retrieval clustering engine which its main purpose is to enable easy performance of experiments comparing different clustering algorithms. It features a strong pipelined component-based architecture which allows researchers to test and verify their algorithms in a tangible environment.

Due to its pipeline-oriented nature, the clustering process is independent of the type of the techniques applied. So researchers can focus on writing the clustering components and then put them together in the process with already existing components. Carrot2

shall be extended, modified or simply run and evaluated by other researchers. Therefore we decided to employ Carrot2 as our clustering search engine instead of other commercially successful alternatives such as Vivisimo [11] or IBoogie [12].

Five clustering algorithms are currently available in Carrot2, among which FuzzyAnts, HAOG and Lingo3G embody the clustering results in hierarchical structure [13]. Lingo3G has the best clustering speed and performance which is a commercial algorithm and is not available in the open source part of Carrot2. Therefore, we have used FuzzyAnts and HAOG algorithms for our purpose.

All of the core components of Carrot2 are written in 100% pure Java, which makes it possible to deploy them on most available software platforms. All of these components communicate via the HTTP protocol and exchange data as XML streams, which gives the ability to physically distribute components and integrate them by means of well-established approaches such as web-services. However due to the low speed of HTTP and XML processing, this shall be considered a trade of between performance and the ease of distribution.

3.2. Prototypical implementation & conceptual schema

The user interface in our prototype is a mixed interface (combination of 2D and 3D) which is more effective than pure 3D interfaces. The 3D interface is displayed in an X3D-enabled browser which permits a user to examine the virtual scene that visualizes search results. The 2D interface contains standard elements such as text fields, buttons and combo box.

In our prototype, a visualization module is employed to transform the Carrot2 query results into the designated 3D metaphor. The 3D output data is based on X3D which is the de facto data format for 3D visualization on the Web [14].

Figure 1 depicts a conceptual schema of our prototype. The integration between the visualization module and the Carrot2 engine is made possible through web services while the visualization module itself applies a model-view-controller (MVC) design pattern [15]. Since there is no well-accepted XML schema available for clustering structures, processing the query results is vendor dependant. This ad hoc approach toward transforming the XML-based Carrot2 query results into the suitable structure to be used by our application was done via JDOM API [16]. Although XSLT technologies such as XALAN [17] might seem to be better solution for doing the job, due to recursive hierarchical structure of clusters, JDOM turned out to be a much better solution for our application.

The visualization module was developed under Apache Struts framework [18]. Apache Struts is an open-source framework which encourages developers to adopt MVC architecture for developing J2EE web applications. Struts is a very well documented, mature and popular framework for building front ends to web-based Java applications and that's why, although there are many newer so called light weight MVC frameworks such as Spring [19] and Tapestry [20] available, Struts was chosen for our implementation.

In addition to employing J2EE technologies on the server side, the most state of art technologies are applied on the client end to maximize portability. In order to take advantage of the 3D visualization, the end user merely needs to use an X3D-enabled browser. All components of the application including Carrot2 components were successfully deployed on Oracle Application Server (OAS) [21].

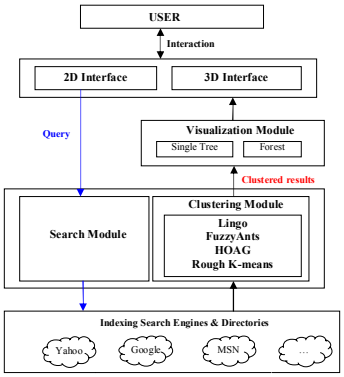
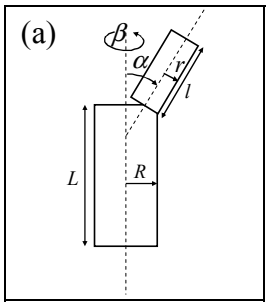


Figure 1. A conceptual schema of our prototype.

3.3. Visualization module

The aim of the visualization module is to transform and visualize the results returned by hierarchical clustering engine into 3D space. To this end, the layout algorithm of the visualization metaphor and metaphor alternatives are described.

3.3.1. Layout algorithm. In our proposed metaphor, a straightforward recursive procedure is used to transform the query result into a tree metaphor where clusters and links are presented as branches (cylinder) and fruits (spheres) respectively. Some parameters with default values shall be altered by user to customize the tree structure. The parameters are defined as follows: L as the length of root branch (trunk), R as the radius of trunk, c as a contraction ratio which shortens the length of branches in deeper levels of the hierarchy



r : Local radius of the sub-branch
l: Local length of the sub-branch relative to its parent branch
β: Local rotational angle of the sub-branch around y-axis in the local coordinate system
translation: Local translation of the sub-branch relative to its parent branch (x, y, z)
rotation : Local rotation of the sub-branch relative to its parent branch (x, y, z, α)

```
PROCEDURE: construct3DtreeMetaphor (parentCluster, L)
    l = L – (L / C)
    FOR each child[i] of the parentCluster
      IF child[i] is a cluster
          β = (2π / number of parentCluster's children ) * i
          r = (R / total number of links) *  number of links in child[i]
          rotation = (Sin(β), 0, Cos(β), α)
          translation = (Sin(β - π/2) * (R - r) ,  L/2 + l/2,  Cos(β - π/2) * (R - r) )
          CALL constructCurrentBranchMetaphor with l, β, r, translation and rotation
                RETURNING currentBranchMetaphor
          CALL construct3DtreeMetaphor with child[i] , l RETURNING sub3DMetaphor
          Append sub3DMetaphor to the currentBranchMetaphor
          Append currentBranchMetaphor to the resultMetaphor
      ELSE
          CALL constructFruitMetaphor with child[i]RETURNING fruitMetaphor
          Append fruitMetaphor to the resultMetaphor
      END IF
    END FOR
    RETURN resultMetaphor
END PROCEDURE
```

(b)

Figure 2. A parent-child status in the tree (a). The recursive procedure which generates the X3D-based metaphor (b).

relative to the parent branch length, α as the rotational angle of the sub-branch against the prolongation of its parent branch (the branch rotate with an angle α around the z-axis in local coordinate system). The branch thickness conveys relative density information of the corresponding cluster. Figure 2(a) depicts a parent-child status of the tree and Figure 2(b) represents the pseudo-code of a recursive procedure which generates the relevant XML-based stream of the 3D metaphor. Figure 3 shows a snapshot of applying algorithm to a simple hierarchy.

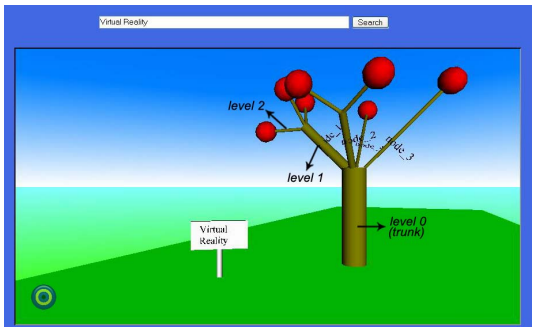


Figure 3. Snapshot of the proposed model for a simple hierarchy.

3.3.2. Single tree or Forest. In order to improve effectiveness and better user understanding of the metaphor, we propose user customizable visualization alternatives according to the way of applying algorithm to the results.

The two alternative metaphors in our prototype are as follows:

- *Single tree*: Mapping entire retrieved results (clusters) to a single tree.
- *Forest*: Considering each parent cluster as a tree and placing them on the ground.

Each of these items is useful for different purposes. Single tree is more suitable for small set of results visualization, however visualizing large clusters in a single tree may clutter the metaphor and confuse the user. Therefore, considering each parent cluster (cluster at the first level of the results hierarchy) as a single tree and placing them on the ground which conveys the forest concept, would allow users to gain better understanding of the metaphor. However the forest metaphor brings a new challenge; growing tree branches on the space may cause collision between two adjacent trees. To solve this problem, we calculate relative position of trees on the ground according to their projection on the xz plane which we call *tree safe area*. To this aim, projection of each tree is considered as a circle with radius of *tree radius*.

Let T be a tree with the *level* branching depth; length of the branches according to their level is defined by:

$$l_{level} = (\frac{c-1}{c})^{level} * L \qquad (1)$$

Where l_{level} represents the branch length in the *level* depth of hierarchy, c represents contraction ratio and L represents length of the trunk. For the trunk *level* is zero.

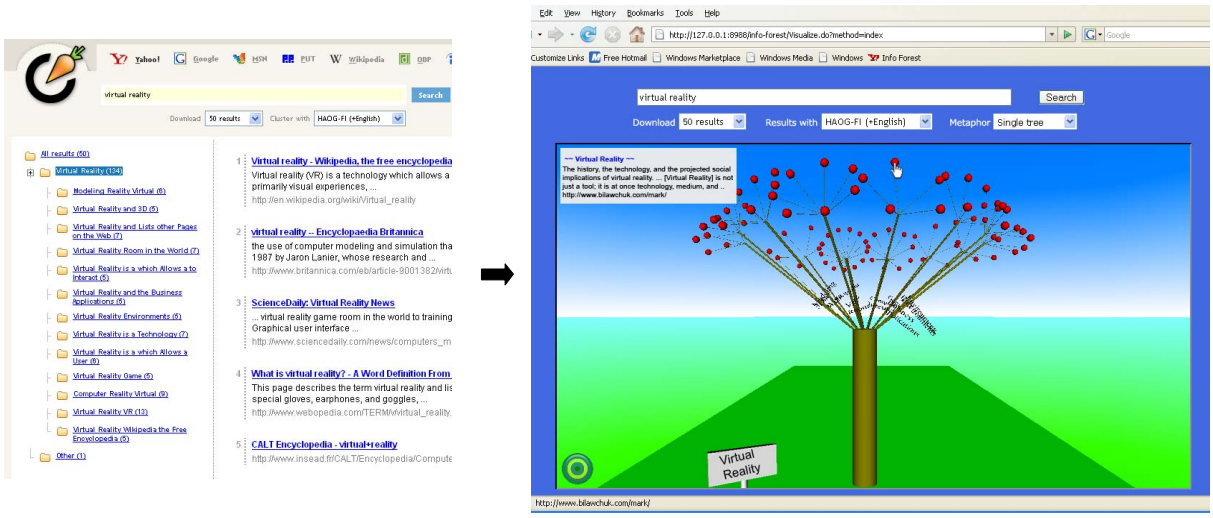


Figure 4. Carrot2 interface of "virtual reality" query term for 50 results and HAOG-FI algorithm (left). 3D visualization of results in the single tree metaphor (right).

Tree radius (d_T) is defined by:

$$d_T = \sin\alpha * [\sum_{i=1}^{level} (\cos^{i-1}(\alpha) * l_i) + 2 * r_f * \cos^{level-1}(\alpha)] + R \quad (2)$$

Where α represents the diversion angle of the branch, r_f represents the fruit radius and R represents radius of the trunk.

Trees are placed on a grid with the following distance:

$$D = 2 * d_{\max} + \delta \quad (3)$$

Where $d_{\max}$ is radius of the tree with maximum *level* of hierarchy and $\delta \succ 0$ which represents user customizable spacing between trees.

5. Application

The proposed visualization metaphor is applied to the results returned by Carrot2 engine. Figure 4 and 5 demonstrate the visualization of 50 results for "virtual reality" query term returned by HAOG-FI and FuzzyAnts algorithms respectively. The proposed mixed user-interface on the right side of the figures is comparable with its Carrot2 interface on the left side. Figure 4 depicts all the results as a single tree. Fruits of the tree represent the last level of hierarchy which is a hypertext to its relative webpage URL. Each branch of the tree is labeled by the corresponding cluster name. By rolling over the fruits, user can see information (URL and snippet) of the relevant document. The clustering algorithms and the metaphors (single tree or forest) are user customizable. Figure 5 shows the results in the forest metaphor. As you can see, the two parent clusters (vr and the world) has been considered as a single tree and are placed on the ground according to their safe area estimated in the previous section.

6. Conclusions and future work

In this paper, we proposed an intuitive metaphor for the 3D visualization of hierarchical clustered results returned by clustering search engines. Our metaphor follows trees as a familiar concept, which is based on traditional node-link structures. We presented a straightforward recursive algorithm which traverses the hierarchy and transforms it into branches and fruits in 3D space. Our model employed Carrot2 as the clustering engine which is an open source framework developed mainly for research purposes. We applied well-accepted J2EE architectural patterns in our implementation model and due to our thin-client policy; the end-user merely has to use an X3D-enabled browser to interact with our 3D representation.

In order to improve the visualization, we incorporated both single-tree and forest metaphors. Therefore in addition to customizing the number of returned results and selecting the clustering algorithm, user will be able to toggle between these two metaphors (single tree or forest). The average running time of the post-processing on clustered data during 3D visualization (while generating the X3D based metaphor) is fairly negligible in comparison with hierarchical clustering algorithms such as FuzzyAnts which usually takes about 2.7 seconds to respond for 100 number of results [13].

We have decided to conduct comprehensive user studies to determine how much the current 3D

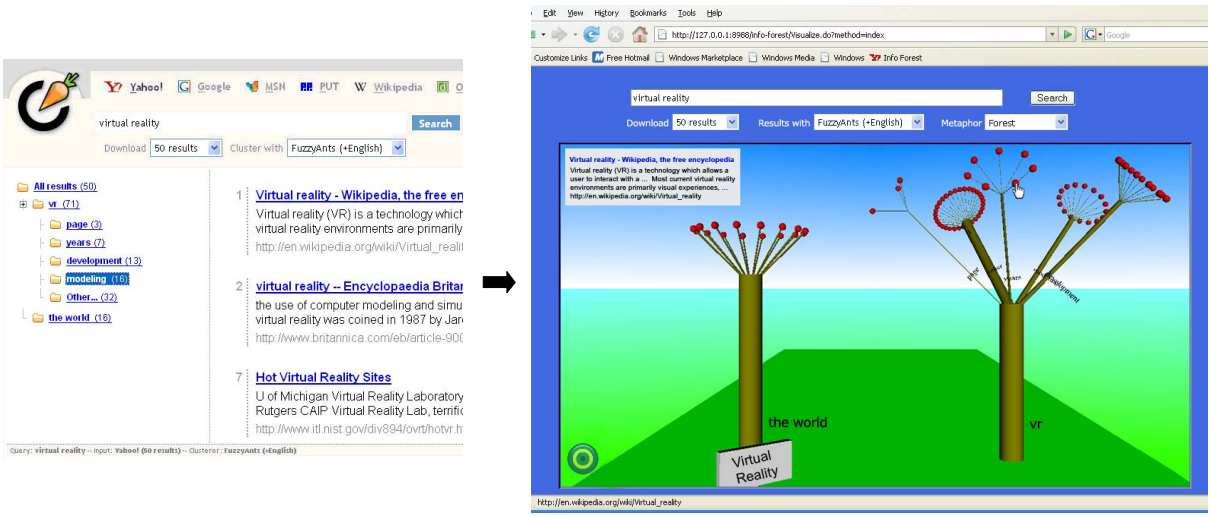


Figure 5. Carrot2 interface of "virtual reality" query term for 50 results and FuzzyAnts algorithm (left). 3D visualization of results in the forest metaphor (right).

metaphor improves the inception of the displayed data. The design and evaluation of other 3D layouts, taking better advantage of 3D space and its rich visual cues, the visualization of greater number of clustered results and supporting more customizable metaphors will be among our future work in this regard. We also plan to design more effective tree layout algorithm based on 3D fractal trees which are applicable to huge hierarchical data structure. However despite the potential advantages, various shortcomings such as difficulty in navigation and complexity of the metaphor must be overcome.

11. References

[1] iProspect, "iProspect Search Engine User behavior study", *White paper*, April 2006.

[2] J.M. Cigarran, A Pe˜nas, J. Gonzalo and F. Verdejo, "Evaluating Hierarchical Clustering of Search Results", Springer-Verlag, Berlin, 2005, pp. 49–54.

[3] O. Zamir and E. Oren, "Visualization of Search Results in Document Retrieval Systems", *General Examination Report*. University of Washington, 1998.

[4] S. Benford, D. Snowdon, C. Greenhalgh, R. Ingram, I. Knox and C. Brown, "VR-VIBE: A Virtual Environment for Co-operative Information Retrieval", *Eurographics'95*, Maastricht, Netherlands, 1995, pp. 349-360.

[5] Kartoo. Available: http://www.kartoo.com/

[6] T. Kohonen, S. Kaski, K. Lagus, J. Saloj¨arvi, J. Honkela, V. Paatero and A. Saarela, "Self Organization of a Massive Document Collection" *Special Issue Neural Networks Data Mining Knowledge Discovery*, IEEE, 2000, pp. 574–585.

[7] N. Bonnel, A. Cotarmanac'h and A. Morin, "Meaning Metaphor for Visualizing Search Results", In *Proceeding 9th international conference Information Visualization(IV'05)*, IEEE, 2005, pp. 467– 472.

[8] W. WIZA, K. WALCZAK and W. CELLARY, "AVE – A Method for 3D Visualization of Search Results", *3rd. International Conference Web Engineering ICWE 2003*, Springer-Verlag, Berlin, 2003, pp. 204-207.

[9] J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur and V. Crow, "Visualizing the non-visual: spatial analysis and interaction with information from text documents". I*n Proceedings Information Visualization symposium*, IEEE, Atlanta, 1995, pp. 51-8.

[10] Carrot2 clustering search engine. Available: http://project.carrot2.org/

[11] Vivisimo. Available: http://vivisimo.com/

[12] IBoogie. Available: http://www.iboogie.com/

[13] Carrot2. Available: http://project.carrot2.org/algorithms.html

[14] Web3D consortium. Available: http://www.web3d.org/x3d/

[15] Model-View-Controller. Available: http://java.sun.com/blueprints/patterns/MVC.html

[16] JDOM API. Available: http://www.jdom.org/

[17] Xalan Component. Available: http://xml.apache.org/xalan-j/

[18] Struts Framework. Available: http://struts.apache.org/

[19] Spring Framework. Available: http://www.springframework.org/

[20] Tapestry Framework. Available: http://jakarta.apache.org/tapestry/

[21] Oracle Application Server. Available: http://www.oracle.com/appserver/index.html