Enhancing Data Interactivity with Data Visualisation Technique on Aakash Tablet

M.Tech. Dissertation

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

By

Sanobar Nishat

Roll No: 113052001

Supervisor:
Prof. D.B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai-400076

Jun, 2013
Dissertation Approval

This dissertation entitled Enhancing Data Interactivity with Data Visualisation Technique on Aakash Tablet by Sanobar Nishat is approved for the degree of Master of Technology.

Examiner-I

Prof. Kavi Arya

Examiner-II

Prof. Avinash Awate

Supervisor

Prof. Deepak B. Phatak

Chairman

Prof. Uday Gaitonde

Date: ————————

Place: ————————
Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____________________________
Signature

____________________________
Name

____________________________
Roll No.

Date: ______________

Place: ______________
Abstract

Information visualization techniques have been studied by many researchers and are developed in many forms for medium to large size display devices. Various innovative information visualisation techniques are developed and evaluated for desktop devices. Increasing demand of hand-held devices (e.g., Tablet PC, PDA (personal digital assistant), smart-phone etc.) necessitates the development of such techniques for these small display devices. Though, these devices pose several limitations in using these visualization techniques on Tablet PC, PDA and mobile phones.

The work illustrates issues and modification in transporting the existing information visualization technique, on small screen devices, particularly on Tablet PCs. An experimental approach is used to compare four well-known visualization techniques including Hyperbolic Tree, Icicle Plot, Space Tree, and Treemap, for displaying hierarchical data. The designs are evaluated on four criteria i) task completion time, ii) task correctness, iii) user interaction and satisfaction, and iv) understanding of design topology. The result suggested that Space Tree visualization is preferred over other visualization for searching tasks while treemap is preferred for topological and comparison tasks.

The work also demonstrates the implementation of a browser-based tool WebVis for visualising hierarchical data. The tool enhances user interactivity with the data and provide a platform to share, explore, and analyse data.

Keywords: Hierarchical data, Hyperbolic Tree, Icicle, Information visualization, Space Tree, Tablet PC, Treemap.
Acknowledgment

I would like to express my deep gratitude to Prof. D.B.Phatak, who has always been making things simple to understand. He has been remarkable in his attempt to keep me motivated in this project and has always provide his feedback and suggestion. I would also like to thank our volunteers for their assistance in conducting experiment. I am also very grateful to all the students who participated in the experiments for their time and feedback. Last but not the least, I would also like to acknowledge enormous help and support of Mr. Nagesh Karmali throughout the project.
## Contents

**Chapter 1  Introduction**  
1.1  Background ........................................ 5

**Chapter 2  Visualization**  
2.1  Process Of Visualization .......................... 7  
  2.1.1  Mapping ........................................ 8  
  2.1.2  Selection ....................................... 8  
  2.1.3  Presentation .................................... 8  
  2.1.4  Interactivity ................................... 8  
  2.1.5  Human Factors .................................. 8  
  2.1.6  Evaluation ...................................... 8  
  2.2  Types of Information ............................... 8  
  2.2.1  Sequential Layout ................................ 9  
  2.2.2  Hierarchical Layout ............................. 9  
  2.3  Classification of Visualization Method ............ 9  
  2.4  Interactivity with Visualization .................. 10  
  2.4.1  Zoomable Interface ............................. 10  
  2.4.2  Zoom + Pan ..................................... 11  
  2.4.3  Overview + Detail View ......................... 11  
  2.4.4  Focus + Context or Fish Eye View ............... 11  
  2.4.5  Rapid Serial Visual Presentation (RSVP) ....... 12  
  2.5  Visualization Techniques .......................... 13  
  2.5.1  The Classical Hierarchical View ................. 14  
  2.5.2  Space tree .................................... 14  
  2.5.3  Hyperbolic Tree ................................ 15  
  2.5.4  Radial Tree .................................... 17  
  2.5.5  Cone Tree ...................................... 17  
  2.5.6  TreeMap ........................................ 19  
  2.5.7  Fluid View ..................................... 22

**Chapter 3  Small Display Devices**  
3.1  Definitions ......................................... 25  
3.2  Limitations of Small Display Devices .............. 26  
3.3  Physical Characteristics of Display Devices ....... 26  
  3.3.1  Display Device Characteristics .................. 28  
3.4  Visual Features ..................................... 28  
  3.4.1  Hue ............................................. 28  
  3.4.2  Luminance ..................................... 29  
  3.4.3  Texture ........................................ 29
List of Figures

2.1 A zoom + pan example [8]. ............................................. 11
2.2 An overview + detail example [5]. ................................. 12
2.3 A Focus + Context or FishEye View example [5]. .......... 12
2.4 An example of the operation of the RSVP technique [7]. .. 13
2.5 The Classical Hierarchical View [10]. .............................. 14
2.6 An example of the Space Tree [10]. .............................. 15
2.7 An example of the hypertree [11]. .................................. 16
2.8 An example of the hypertree, changing the focus [11]. .. 16
2.9 An example of the Radial tree [14]. ............................... 18
2.10 An example of the detailed view of Radial tree [14]. .... 18
2.11 An example of Cone tree [5][15]. ............................... 19
2.12 A ConeTree displays category labels and a WebBook shows retrieval results. The left-hand page shows the title and the category labels associated with the document. The right-hand page shows the abstract associated with the document [15]. ............................................. 20
2.13 An example of Tree Map [? ]. ..................................... 21
2.14 Fluid Views comprise two interconnected layers: the top layer features items at varying scales superimposed over a zoom-able base layer, for example, representing time, location, and content [19]. .............................. 22
2.15 An example of Fluid View [19]. ................................... 22
2.16 To see the exact positions of cluster members, a cluster can be opened following a transition arranging the items around the cluster and displaying edges to their actual place on the base map [19]. .............................. 23
2.17 Image on left-hand side: Zooming from a continental view (a) to the Mediterranean (b), and finally a regional close-up (c). With increasing geographical detail in the map, the displayed blog articles are continuously refined. Image on Right-hand side: The time scale is changed from (a) two months, to (b) two weeks, and (c) seven days [19]. .............................. 24
3.1 Visualization of a weather dataset using perceptual texture elements with temperature hue, wind speed density, pressure size, precipitation orientation, and cloud coverage luminance [23] ......................... 29
3.2 Visualization of a weather dataset using perceptual texture elements with temperature hue, wind speed density, pressure size, precipitation orientation, and cloud coverage luminance [23] ......................... 34
4.1 User logarithmic Response time in seconds for each task on following four views: Hyperbolic Tree (HT), Icicle Plot (IP), Space Tree (ST), Treemap (TM). Response time includes correct and incorrect response. .................. 42
4.2 Percentage of Correct Responses for each task on following four views:
Hyperbolic Tree (HT), Icicle Plot (IP), Space Tree (ST), Treemap (TM).

5.1 An Interactive Slider View

5.2 A View of Hierarchical Data in Tabular Format

5.3 JSON file format for Hierarchical Data

5.4 A View of External File Uploading Interface

5.5 A View of User Interface for Accessing Database with Query Generation
and Advance Search Functionality

5.6 Search User Entered Node in the Tree

5.7 Treemap: Full View of Tree Structure

5.8 Treemap: Collapsible View of Tree Structure

5.9 Treemap: Collapsible View of Tree Structure without Nodes’ Details

5.10 Pop-up Window for Displaying Files/Web-Page
Chapter 1

Introduction

1.1 Background

Visualisation, as term suggests is a technique to communicate a message through pictures, graphs, diagrams, and animation. Visualization provides users a quick and better understanding of data. It helps in exploring the structure of data and explains the relationship among data. The representation of abstract information in visual form, supports users in solving various tasks and in taking better decisions by unveiling the patterns and trends in the data [1]. The technique has touched almost all domain of computer application with the rapid increase of technology, and graphical power of computers.

This success provides an opportunity and a huge demand to implement this technology on small devices like PDA (Personal Digital Assistant), smart phone and tablet PC. Unfortunately, the transportation of these techniques from desktop computers to small display devices is not a straight forward process due to various limitations of these devices [2][3].

- Limited display screen size.
- Different aspect ratio from the usual 4:3 e.g. 16:10, 16:9 as shown in Table 1.1.
- Limited computational power, limited hardware(CPU, memory, buses, graphic hardware).
- Human interaction techniques (e.g. small keypads, digitizer pen, touch) are often inadequate for complex tasks.

Besides all these limitations, the job of a good visualization technique, is to minimize user effort, and to provide a smooth exploration of information. An efficient visualization method can display a large amount of information in a small space. It helps in reducing complexity, and increases understanding of data [4].

This work mainly deals with the usage and evaluation of visualisation technique for hierarchical data. Significant work has been done in research community in providing innovative visualising techniques for relation and tree structured data. However, all these contributions are done for medium to large size display screen. The evaluation of these techniques has also been discussed in few literatures. There is no significant work done that demonstrates the usage and evaluation of these techniques on small display devices. In this work, an effort is made to provide a comparative study of four
Table 1.1: Device display features

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Model</th>
<th>Screen Size (Diagonal inch.)</th>
<th>Resolution</th>
<th>Aspect Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datawind1</td>
<td>Ubislate 7Ci,</td>
<td>7.0-inch</td>
<td>800×480</td>
<td>16:9</td>
</tr>
<tr>
<td></td>
<td>Ubislate 7C+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samsung2</td>
<td>Galaxy TAB2</td>
<td>7.0-inch</td>
<td>1024×600</td>
<td>16:9</td>
</tr>
<tr>
<td>HTC3</td>
<td>Flyer</td>
<td>7.0-inch</td>
<td>1024×600</td>
<td>16:9</td>
</tr>
<tr>
<td>Apple4</td>
<td>iPad 2</td>
<td>9.7-inch</td>
<td>1024×768</td>
<td>4:3</td>
</tr>
<tr>
<td>Hp5</td>
<td>ElitePad 900</td>
<td>10.1-inch</td>
<td>1280×800</td>
<td>16:10</td>
</tr>
<tr>
<td>Lenovo6</td>
<td>ThinkPad Tablet 2</td>
<td>10.1-inch</td>
<td>1366×768</td>
<td>16:9</td>
</tr>
<tr>
<td>Sony7</td>
<td>Xperia Tablet Z</td>
<td>10.1-inch</td>
<td>1920×1200</td>
<td>16:10</td>
</tr>
<tr>
<td>Samsung8</td>
<td>Series 7 Slate</td>
<td>11.6-inch</td>
<td>1366×768</td>
<td>16:9</td>
</tr>
</tbody>
</table>

different visualisation layout: (i) Space Tree, (ii) Hyperbolic Tree, (iii) Icicle Plot, and (iv) Treemap, for displaying hierarchical data on Table PC. Few limitations of using these layout on such devices, are also discussed.

The report is split into six chapters. Chapter I presents a brief introduction of the motivation of my work. In chapter II, a detailed survey is presented on different visualisation techniques and interaction methods with these techniques. The effects of device capabilities and their features on data visualisation is thoroughly explained in chapter III. It also covers the limitations and issues with small screen devices that hinders in the effective data visualisation process. In chapter IV, a comparative study is presented of using four common visualisation method on Table PC. Chapter V demonstrates the implementation of a browser-based data visualisation tool for viewing and exploring data in a more interactive way.
Chapter 2

Visualization

“If I can’t picture it, I can’t understand it.” —Albert Einstein

Visualization takes in the pictorial representation in your mind what is happening in the text. Visualization of narrative text makes use of sensory images like sounds, touch, physical sensations, smells, and emotions communicated in the story to help you picture the story. The most referred definition in literature explains Visualization as: “the use of computer-supported, interactive, visual representations of data to amplify cognition”, where cognition is the neural system’s function to perceive, record, process, and use the information gathered by sensory receptors.

The question arises that why is visualizing so important? The answer is because the brain “sees” in order to save and process information. Visualization provides the transformation of a non visual quantifiable data into a visual representation form for brain to better understand and explore the data and its hidden patterns. The soul purpose that visualization fulfills is to figure out what is insight. Visualization is use to explore, illustrate, discover and convey information in well explanatory form. Visualization is efficient in presenting bulk of data compactly and coherently and furnishes multiple layer of details from different view points and provide them on-demand. Besides having various benefits of visualization, it is equally essential to evaluate it in the context of its usage, effectiveness, ease of incorporating them, and robustness. Visualization is a powerful technique for performing various cognitive specific process like descriptive, exploratory and analytical [5].

2.1 Process Of Visualization

The process of constructing a well disciplined visualization consists of different steps, briefly lists in the literature of “Chittaro, 2006”. The list comprises of the following different steps i.e. (i) mapping, (ii) selection, (iii) presentation, (iv) interactivity, (v) usability, and (vi) evaluation, which describes the major task involving visualization and to give accurate and error free model. Explanation of these steps is given in the following sub-sections [2][4].
2.1.1 Mapping
Mapping is the first act towards achieving visualization. It is the process of analyzing the data and encode them in to visual form. Through this process, data or information transforms in to a graphical representation based on given visual features. Depending on the clarity and concreteness of the relationship between data objects and visual objects, a more accurate and precise visual form can be generated through mapping.

2.1.2 Selection
Selection is the second action in the visualization process. Selection means to choose the appropriate data among the mass of data suited for given task or job. The process aims to the selection of data pertinent for the visual graphics or pictorial representation. In the visualization process, the selection step considers to be the most crucial as the selection of wrong data can mislead the user while taking major decision; hence great care needs to be taken in the data selection process and should refrain from unnecessary data.

2.1.3 Presentation
Presentation is the third step for visualization. In the context of visualization, presentation means how to effectively convey the textual information in the pictorial form in the limited available screen space. After affective mapping, careful selection of data items it is equally important to present it in more concise, understandable and useful form.

2.1.4 Interactivity
Interactivity is the fourth step in the visualization process. In this step, the main focus is to provide an easy and user friendly interaction considering the provided facilities to explore, establish and rearrange the visualization. A good user interactivity enables a user to easily analyze, understand, and annotate the data or information, by enhancing their exploration capabilities.

2.1.5 Human Factors
Human factor is also an important consideration for designing visualization feature. These factors categories into two broad group, usability and accessibility. End user should find easy to use the visualisation. The use of human visual perception and cognitive knowledge is very effective in designing of the visualization and a common practice in Human Computer Interaction.

2.1.6 Evaluation
The most intrinsic task in visualization is to evaluate it for the best use depending upon the provided features and limitations. Evaluation is also needed to examine the effectiveness of visualization method suited for the application or Information, and that the visualization accomplish the goal or not.

2.2 Types of Information
With the increasing amount of information and complexity among their relations, efficient techniques to layout the data on the display screen, are being more and more
important. Based on the types of Information, layout can be broadly divided into two forms [3].

1. Sequential layout
2. Hierarchical layout

2.2.1 Sequential Layout

The information that are linear or has less interrelation between data can be efficiently display using sequential layout. For example, the linear list of websites as a result of a query search on search engines i.e. Google. This method doesn’t require to interpret data in some other visual form e.g. graph, chart. It simply displays the results in the form of the list.

2.2.2 Hierarchical Layout

Hierarchal layout effectively captures the complex relation and hierarchy between information. Through this arrangement, complications in the data can be simplified in to simpler form which also reveals hidden trends and patterns in the data. Horizontal, vertical and circular or radial layout are different formats to represent Hierarchal information. Different visualization techniques uses different layout matches to the specific needs. Tree, charts and graphs are the traditional approaches used in hierarchical layout.

In the subsequent section, different visualization techniques and layouts are discussed in detail for sequential and hierarchical types of information.

2.3 Classification of Visualization Method

A variety of Visualization techniques has enabled its implication in different domain and application. Based on that, different visualization methods and techniques are categorized differently by different author. Broadly, the term visualization divided in to three main category i.e. (i) Information Visualization, (ii) Scientific Visualization and (iii) Software Visualization [2].

Information Visualization: The illustration of information by using graphical and spatial representation in order to explore and understand data and to facilitate comparison, change detection, pattern recognition. There are other sub categories of visualization mentioned in different literature i.e.

- **Data Visualization** represents the data visually in diagrammatic or schematic forms. Different techniques e.g. Table, Line chart, Histogram, Scatter plot, and Pie chart etc. are used in Data visualization.

- **Information visualization** facilitates the user to enhance their cognition or perception capability through the interactive interface. Dynamically created visualization helps the user to better manipulate and understand the data, e.g. Data map, Semantic network, Time line, and Venn/Euler diagram etc.

- **Concept visualizations** helps in elaborating concepts, plan ideas and provide an easy mechanism to analyze them e.g. Concentric circle, Mindmap, Decision tree, Layer chart,
Pert chart etc.

Strategic visualization deals with the strategical behavior and decisions of an organization. It provides a way to visually represent the development, formulation, communication, implementation and analysis phases in the form of strategy map, Organizational chart, Portfolio diagram, and Failure tree etc.

Metaphor visualization communicates the insight of the information using the graphical structure assuming the key characteristics of metaphor that is applicable, e.g. Metro map, Funnel, Story template etc. Compound visualization makes use of different graphic representation format fitting on a single schema or frame e.g. Knowledge map, Cartoon, Rich picture, and Learning map etc.

Scientific Visualization: In order to understand the physical phenomena in data, mathematical models, volume rendering, and glyphs etc., scientific visualization provides different techniques and interfaces to efficiently explains scientific data and model. Researchers have made distinction between information visualization and scientific visualization based on the analysis, that the former uses non-numeric data such as text and graph while latter uses numerical data. But this distinction doesn’t hold always in practice. Many Information visualization applications use numerical and textual data to provide more concreteness to the visualization.

Software Visualization: Software visualization aids people to learn and acquire the skills of using computer software. Program visualization benefits programmers in handling complex software, and similarly provides support in algorithm animation, foster, and motivate student to understand the computation capability of an algorithm.

In this work, we will be focusing on the two widely use visualization methods from the above discussed techniques i.e. Information and Data Visualization.

2.4 Interactivity with Visualization

Through the process of visualization, an abstract data is extracted from the huge raw data. So, it is highly desirable to provide an easy and efficient interactive visualization techniques to explore, understand, analyze the information presented in the visualization. One of the challenge in the visualization is to present only the relevant data in such a visual form which is understandable to the user with minimal effort. For this purpose, researchers proposed various interactive techniques that makes the manipulation of visualization efficiently and effortlessly. User has different ways of interaction with the visualization or interfaces e.g. Mouse single or double click, mouse over, mouse right or left click etc. Card et al in 1999 presented the human interactive mechanism of visualization consisted of different phases as mentioned below:

2.4.1 Zoomable Interface

Zooming is an important technique in computing which allows user to interact with the interface and see the required information much closer and in much detail. Zoomable user interface “ZUI” in graphical environment facilitates the user to set the scale of the viewing space at varying level of details. User can use the zoom in technique when a detail view is needed and gradually zoom out to see a larger view. This technique is
best suited for device having limited display size in comparison with the amount of data to be shown. Different strategies are proposed to provide different level of views, from abstract view to detailed view [5].

2.4.2 Zoom + Pan

The Zoom + Pan technique is a collaboration of two separate technique i.e. zooming and panning to access the user interface. Through the use of zoomable technique, a user can drill down to some extent of lower level details of interface while can also zoomed out to have an overview of the interface. Through the use of panning technique, user is able to move across the content of the viewing area. The panning technique works in both zooming conditions i.e. zoom in and zoom out. The two important benefits of using this strategy are the effective usability of the screen size and provision for scalability. Some of the disadvantages of panning technique is the loss of overview and slower navigation [5].

![Figure 2.1: A zoom + pan example](image)

2.4.3 Overview + Detail View

In the overview + detail view technique, user interface is provided with multiple views of the concerned information at the same time. One smaller view locates the detail view within a larger view and also provides the overview of the information space. In the other larger view the located information of the portion of the information space is displayed in much detailed view. These technique is very common in map navigation, picture or image browsing software. The degree of zooming from detail view to overview for 2D images is 30:1, given the limitation of usability to achieve the desired details for navigation [1][2].

Through the method of conserving the overview with the detail view facilitates the user from disorientation in the detail view, but causes visual breach because of switching between the the two views. One of the advantages of this technique is the availability of multiple view or chained view or scalable views. But this technique is not very effective for very small screen display because of the presence of multiple view [1][2].

2.4.4 Focus + Context or Fish Eye View

Focus + Context or Fish Eye view makes use of the overview + detail techniques with some different interaction strategy. In this method all the available space is utilized
to show the information in the overview context. While a focus area on the overview space is magnified to show the details of information for portion of the information in focus. In order to keep the local and global context together on the same view, the surrounding overview slightly descents by wrapping the overview. Due to the above mentioned technique, it is named as fisheye view or distortion- oriented display. The main advantage of this technique is keeping the context when focus i.e. show the surrounding of the focus area to provide an easy navigation for the user. Distortion and unstable overview are its main disadvantages. Figure 2.3 shows the focus + context technique [2] [5] [6].

2.4.5 Rapid Serial Visual Presentation (RSVP)

Rapid serial visual presentation (RSVP) is Information presentation technique, mainly suited for displaying text data. In the RSVP technique, the text is presented in chunks rapidly in a fixed focal position for a fixed interval of time suitable for the user to search through the Information. Electronic RSVP is more like an electronic riffling through the pages of a book. Video-on-demand browsing systems, shopping service, and video browsing using dynamic key frame presentation advocate the successful use of RSVP within these context. Some studies also supported the use of RSVP technique as a result of the enhancement in the user information processing speed in comparison with the simultaneous presentation of complete text [2].
RSVP seems faster than MS Explorer for rendering smaller as well as longer text on a PDA device. RSVP method is regarded as efficient as reading from a book or large screen, but it demands user attention and alertness [2].

Visualizing large pictures on a limited space can be done efficiently with the use of RSVP technique by displaying the sequence of regions of a picture in cycle. An automatic method for the identification of the likely user’s region of interest is used for the selection of the region. The RSVP technique lowers the user effort of zooming and scrolling [2] [7].

**RSVP Browsing** is an interaction model designed to support both link traversal and backtracking using RSVP technique. RSVP browsing is proposed to support the presentation of links, which can be followed during Web search, and page trajectories in case the user wants to revisit previously visited pages. It consists of controls necessary for starting and stopping RSVP, and selecting links and previously visited pages [8].

### 2.5 Visualization Techniques

Researcher proposes various layout for the visualization of the hierarchal or tree like structure which broadly can be classified into two main streams:

1. Connection
2. Enclosure

#### 1. Connection

It is the most conventional representation of the hierarchal data in the form of node-link pair structure. A set of nodes representing Information are drawn in the diagram and a set of edges (link) showing the connections or relationship between the nodes i.e. parent-child relationship. The benefit of using node-link diagram to describe hierarchal data is most appreciated in the perspective of human visual ability to deduce the relationship in the graph. Many researchers have been done in this direction such as cone tree, hyperbolic tree, radial view etc [9].
2. Enclosure

Another approach to visualize tree structure is to use enclosure. Enclosure method uses a bounded area to represent the data. In this method each node of the tree is represented as a rectangular space, which is followed by subdivision of the space in horizontal and vertical direction to map the relative size of the children of the node. This is a recursive process which iteratively sub-divides the children node space into rectangular area to represent their children. Some models have been proposed by the researchers in this direction. An example of this type of visualization is tree maps. Node-link structure and tree maps are both valuable techniques for tree visualization of fairly large size of data. But efficiency of each one varies with the properties of data. Node-link diagram is most suited for the tree structure of uneven shape while tree-maps approach is most effective to represent quantitative or numbered data [9].

The following sections provide insight of each of these techniques briefly and explains the limitations of using these techniques with different datatypes [9].

2.5.1 The Classical Hierarchical View

The traditional hierarchal view of the tree is based on the algorithm established by Reingold and Tilford. In this view, the children nodes are positioned below their common predecessor. The tree view can grow vertically as top-down, horizontally as left-to-right or grid like positioning. Some of the advantages of this algorithm is its simple computation and faster speed. But a lot of unused space gets wasted because the algorithm is not optimal in using the geometric space efficiently. Therefore, this technique is not suitable for large size hierarchal data visualization [9].

![The Classical Hierarchical View](image)

2.5.2 Space tree

Space tree, a novel design that incorporates the zooming environment with the conventional node-link tree layout that dynamically expand the branches of the tree to best fit the available display space illustrated in Figure 2.6. The interface uses preview icons to encapsulate the topology of the branches when the space isn’t enough to show them at once. When the user clicks on the preview icon or the corresponding node to navigate among siblings, ancestors and descendants, the focus of the layout sets accordingly. Figure 2.6 demonstrates the how SpaceTree accommodates the lower level of hierarchy...
to be opened in the interface. It is observed that "Semantic zooming is preferred over geometric scaling" which means that if the detail of the node of lower levels of the tree is not visible then show them in an aggregated or abstract representation. The consistent layout of SpaceTree helps user to quickly return to already visited nodes provides user an appropriate interface for the frequent performing task [10].

![Space Tree Diagram](image)

**Figure 2.6: An example of the Space Tree [10].**

This technique is suited for nominal size tree visualization. Space tree visualization shows benefit for navigation tasks and provides an easy track to already previously visited nodes. It also helps in estimation of the overall tree topology. It is more appropriate for the representation of textual data.

**Suitable Datatypes:** Suitable Datatypes for space tree visualization technique are hierarchical information structures having parent-child relationship.

### 2.5.3 Hyperbolic Tree

A hyperbolic tree or hypertext uses hyperbolic geometry technique to depict data in a graphical drawing technique. The soul purpose of using the hyperbolic tree is to minimize the cluttering of nodes and efficient use of space which happens in the case of displaying data as a tree structure. Because the number of nodes increases exponentially for larger trees and requires exponential amount of space to display the data in node-link diagram. The hypertext design, illustrated in Figure 2.7, was at first proposed by the Escher woodcut. Two salient features of the techniques are: first the displaying strategy and second that the ability of the design to accommodate the exponential growth in the number of components. The display space given to a node begins to diminish as a continuous function with the increase of its distance in the tree from the focus of attention. The hypertext uses the power of focus+context or fisheye techniques well suited for tree visualization to smoothly transit between focus and detail view [11][12].
Hyperbolic Technique  The gist of this approach is to arrange the hierarchical components on the hyperbolic plane and then display this mathematical formulation of the circular display region. Unlike Euclidian geometry, in hyperbolic plane the parallel lines representing the edges diverge from each other. The property of circumference of the circle i.e. it grows exponentially with its radius, is effectively utilized by the hyperbolic structure which means more space is available to display nodes as the distance increases from the root. Hence hierarchy, that expands exponentially with the depth, can be distributed uniformly in a hyperbolic space, designates equal distance between parents, children, and sibling across the hierarchy. Consequently allows appending of several generations of parents, siblings, and children, aids in easier traversing through the hierarchy for the user without getting lost. In comparison with the conventional hierarchical technique, hypertext is capable of displaying up to 100 times as many nodes and a better means of navigation across the hierarchy [12].

Figure 2.7: An example of the hypertree [11].

Figure 2.8: An example of the hypertree, changing the focus [11].
Changing Focus  The mapping is done in such a way that portions of the plane near the origin uses more space in comparison with the other portions of the plane. Very distant parts of the hyperbolic plane receive minuscule amount of space near the boundary of the disk. Using the focus+context method, the translation of the hyperbolic plane structure maintains the hierarchical geometric model as captured in Figure 2.8. The translation mechanism efficiently control the distribution of space for different portions of plane without compromising the visibility of the detail view [12]. The user can interact with the hypertree by clicking on any visible point to bring it into center of focus or by dragging to any other position. The display transforms accordingly in either case by magnifying the regions approaching the center while shrink as they are progressing towards the boundary [11].

Hyperbolic Tree is best suited for presenting a large hierarchal data. Tasks require navigation over named nodes can be easily represented using hyperbolic tree. The layout is capable of providing user to visualize a group of associations [12]. The hyperbolic tree view helps in identifying visual clues for the distribution of entity types and the ease of comparing each required item to the benchmark at the same time.

Suitable Datatypes:  Hyperbolic tree visualization technique is capable of displaying a very large hierarchical information structures. It can effectively display very large textual data tree of about thousand nodes with the depth of the tree of 5 or 6 level while it can also display visual resource of nominal size efficiently [11].

2.5.4  Radial Tree

A radial tree, or radial map is another efficient displaying technique for a large tree structure in a limited space. The center of the region under visual focus represents the root node and the lower level nodes expand outward radially from the higher level nodes as shown in Figure 2.9. Layouts can be initiated by working outward from the center, root node. The first level nodes can be distributed evenly because of having same parents. In succeeding levels, the nodes are arranged in the remaining space of the assigned sector efficiently to avoid overlapping of the children nodes of one parent. A fixed distance is maintained between different levels of hierarchy to efficiently fit the graph in the screen. The radial tree has some basic similarities to a hyperbolic tree. The primary difference between these two layouts is that hyperbolic trees are designed using the hyperbolic geometry, whereas in a radial tree follows a linear geometry between the orbits of the circular region [5][13].

Sometimes, it is useful to be able to visualize the complete hierarchy all in one go. One of the most efficient ways of doing this is to use a radial tree view. It works in much the similar way as Hyperbolic Tree but uses linear geometry.

Suitable Datatypes:  It works effectively with the displaying of a large textual and visual data. As mentioned in the above section, Radial tree is also able to render about thousand of nodes of multiple level of hierarchy efficiently.

2.5.5  Cone Tree

ConeTree is a novel 3-dimensional user interface with the root node at the top and the first level nodes are dangled down in a circular shape connected to the root node by links depicts a cone-shape structure depicted in Figure 2.11. The nodes at the lower
Figure 2.9: An example of the Radial tree [14].

Figure 2.10: An example of the detailed view of Radial tree [14].
lever also lays out in cones all the way down to the leaf node at the bottom of the screen. The 3-dimension layout of the tree and the animated rotation of the cones to bring the neighborhood nodes in the for-ground is of high interest for the user for eye-catching presentation. The 3D hierarchy layout is very effective in maximizing the use of available screen space and provides the mean of visualizing the whole structure. While the use of interactive animated makes use of human perceptual system and aids the user to easily navigates the tree.

Initially the tree shows all the top level categories and provides an starting point to the users. The users further explore the subsequent node based on their requirement. User selects the node to expand in detail, which displays all of its descendants nodes. The out of scope nodes can also be pruned to provide a smaller view space [5].

Figure 2.11: An example of Cone tree [5][15].

Cat-a-Cone is a technique proposed by Hearst and Karadi to represent a very large hierarchal data using the Cone Tree layout. It integrates the browsing and searching technique with the ConeTree and presents the complete hierarchy in one window while unveil the occluded portions through animation [15].

Cone Tree technique is in huge demand for an appealing 3-dimensional representation layout. Textual as well as visual resources can be represented through this visualization. It is suited for a medium size tree structure.

**Suitable Datatypes:** Suitable Datatypes for Cone tree visualization technique are hierarchical information structures having parent-child relationship.

### 2.5.6 TreeMap

Treemap is one of a novel technique to represent hierarchical data as a collection of nested rectangles shown in Figure 2.13. Treemap is a space limiting and screen filling algorithm which efficiently utilizes all the available space. Each branch of the tree is
Figure 2.12: A ConeTree displays category labels and a WebBook shows retrieval results. The left-hand page shows the title and the category labels associated with the document. The right-hand page shows the abstract associated with the document [15].
represented by a rectangle, which further tiled with smaller rectangles representing sub-branches. The attribute of the data describes the size of each leaf rectangle and the size of interior rectangles is calculated as the sum of the attribute values of its subtrees while color of the node depicts any other attribute of the data [16]. Many algorithm have been proposed by researchers provides further refinement to the TreeMap representation. Slice-and-dice is the original recursive algorithm that arrange the data in some meaningful order (alphabetical, chronological, etc.) in TreeMap, but resulted in a bar-shape layout. "Squarified" algorithm proposed by M. Bruls is a widely use TreeMap layout. In this design the large squares representing the root node are placed in the upper left and small squares in the lower right, provides ease of selection and often allow better labeling [5][13].

![Tree Map Example](image)

**Figure 2.13: An example of Tree Map.**

This visualization technique is most suited for representing quantitative data with multidimensional attribute. It is extremely useful and capable of portraying several different sorts of information in one diagram at the same time. The size of each rectangular region represents the numerical value of data and different attributes of data is captured using rectangles of different colors. The use of TreeMap is most feasible to represent textual data. Treemaps are known to be weak for traversal and nesting levels tasks [17]. Moreover the difference in the orientation in the Treemap makes the comparison task difficult and slower. It is observed that user finds difficulty in comparing rectangular areas of Treemap if both the vertical or horizontal dimension of the two rectangles are different [18].

**Suitable Datatypes:** Suitable Datatypes for TreeMap visualization technique are hierarchical information structures having quantitative attribute of data.
2.5.7 Fluid View

Fluid View is another information visualization technique that aims to reveal patterns and relationships in large dataset. It is a search interface concept that integrates two layers i.e. visual background called as base layer and item layer, representing items at varying levels of detail on top of a data facet as explained in Figure 2.14. The main goal is to provide ease of transition from overview to detailed outlook, display information in context of their collection, scale results based on their relevance or popularity, and provide user interaction to the diagram [19]. Figure 2.15 illustrates the different layer of Fluid view visualization.

Figure 2.14: Fluid Views comprise two interconnected layers: the top layer features items at varying scales superimposed over a zoom-able base layer, for example, representing time, location, and content [19].

Figure 2.15: An example of Fluid View [19].

The base layer provides visual landmarks in the zooming environment, positional
meaning and interpretation for the items, and repositioning ambiance within the zoom space. The item layer is used to display the information items for the viewer’s attention. Search queries and direct manipulation techniques are also embedded in the design. Semantic zooming is incorporated that gradually varies the scale of the display item with sensible increases of detail. When the size of the visual resource is very small, it is represented by a square filled with the dominant color of the image and gradually replaces by the actual thumbnail as the size increases. Similarly when the size of the textual resource reaches a certain size, the whole document can be visualized within the interface [19].

When many items are in close proximity then circular cluster technique is used to group the relevant items together to avoid overlapping. The items within the cluster can be expanded by clicking on it as depicted in Figure 2.16. The items extents towards the boundary of the cluster. The position and color of a cluster is calculated by the weighted average of the positions and average of the colors of the enclosed item. The size of the cluster signifies the value of the data [19]. Figure 2.17 shows the results of two searching scenarios using two different base layer i.e. time-line and map with different level of details as the user navigates through the layout.

![Figure 2.16](image)

Figure 2.16: *To see the exact positions of cluster members, a cluster can be opened following a transition arranging the items around the cluster and displaying edges to their actual place on the base map [19].*

Fluid Views visualization can be summarized by a 6-step process [19]:

1. Positioning: set dimensions and extent of base map.
2. Selection: determine which items are in view using position information from the base map.
3. Ranking: calculate relevance values based on item popularity or search matches.
4. Spacing: allocate screen space and set item sizes.
5. Clustering: group items based on their proximity.
6. Draw: display, change, or hide items and clusters.

All the above mentioned 6-steps triggers with each panning and zooming operations of the base map while searching triggers the last four steps. Only the display get changed by detail-on-demand operation i.e. Step 6. [19].
Figure 2.17: Image on left-hand side: Zooming from a continental view (a) to the Mediterranean (b), and finally a regional close-up (c). With increasing geographical detail in the map, the displayed blog articles are continuously refined. Image on Right-hand side: The time scale is changed from (a) two months, to (b) two weeks, and (c) seven days [19].
Chapter 3

Small Display Devices

This chapter presents and discusses the various available small display devices and establish a backdrop for the evaluation of visualization techniques [20].

3.1 Definitions

Different Information visualization techniques can be effectively evaluated in the context of small display devices, when one understands what is meant by small displays. The screen resolution is the parameter to measure the size of the display. Resolution can be explained as the number of pixels (color points) accumulated on a display element. It is the number of pixels on the horizontal axis and vertical axis e.g. the resolution of hand-held device is $96 \times 65$ pixels.

Sharpness of the image also depends upon the resolution and the physical dimension of the display element. The image will look sharper on smaller display while gradually decrease its sharpness on larger screen for the same resolution because of the spreading out of the same amount of pixels over a larger space.

The standard CRT desktop displays presents resolution upwards to $1600 \times 1200$ pixels and more. While a typical hand-held devices including PDA (Personal digital Assistant), smartphones usually adopt LCD (liquid crystal display) technology. The resolution of display of these handheld devices ranges typically from $96 \times 65$ pixels to $640 \times 240$ pixels.

When considering the effective size of a display, another parameter i.e. aspect ratio should also be considered along with the resolution. The individual pieces of information presents on the display based on the measure of display resolution, whereas aspect ratio indicates the shape of the display.

The effect of aspect ratio will be explained further below considering a real world example. A street map application running on $75 \times 130$ pixels resolution display will work better than $10 \times 1000$ pixels resolution display, although the amount of pixels in both the display are same. In reality the situation is much more complex, where the types of data being visualized have also given a significant consideration in deciding for the optimal shape of the display. There are other additional factors that contributes to effective display size including the capability of the display element to reproduce colors.
3.2 Limitations of Small Display Devices

Adaptation of visualization techniques on small screen devices directly from desktop computers is impossible due to fundamental limitation of these devices. Additionally, the mobility context is different from conventional visualization in many ways. For developing visualization applications for small display devices, some differences should be taken into account [20][21]:

- Limitation in the size, resolution colours of the display.
- The Width/Height ratio is different from the 4:3 which is common in small devices.
- Limited Computational power, limited hardware (CPU, memory, buses, graphic hardware).
- Human interaction techniques (e.g. rollers, micro-joysticks, tiny keypads) are not adequate for complex operations.
- There is a difference in input techniques, e.g. pattern recognition and hand-writing on small surfaces, point-and-tap with stylus, one-hand thumb-based input.
- Bandwidth and Connectivity issues affects the interactivity of application significantly for large data.
- High-level graphics library are not present.

3.3 Physical Characteristics of Display Devices

This section captures the different physical characteristics of display device and visual abilities of the viewer to best match with the visualization design [20].

Different Data attributes could be accurately analyzed and explored using various visual features in the form of images and graphs. Visualization takes advantage of the human low-level visual system to perceive the provided knowledge. In order to take the full advantage of the visual bandwidth of the human visual system, certain criteria needs to be considered:

1. Display device’s physical characteristics (e.g., resolution in terms of the physical size of the display, and the total number of pixels).
2. Accuracy of the human visual system (e.g., the distinguish ability limitation of the human eye towards different visual features for example size and orientation, color etc).
3. Visualization technique (e.g., the approach used to map a data elements values to a visual representation).
4. Nature of the data (e.g., its dimensionality and number of elements) and the analysis technique to be performed by the viewer.

A lot of research has been done on the visualization techniques and properties of data but very few work has been done on the first two criteria, i.e., how visualization can be enhanced by understanding visual features and visual acuity and how they effect
visualization. The foundation for these studies can be provided by the knowledge from human psychophysics and computer vision [22].

The different visualization algorithm developed to date assumes that the display resolution is sufficient to produce visualization effectively. Rapid increase in technology of small display devices has increased the necessity to select the appropriate visualization algorithm for these devices too. Display devices characteristics, can have a considerable effect on a particular visualization technique. These characteristics i.e physical size, standard viewing distance, pixel resolution, varies across different display devices.

Human visual system is another factor that also needs to be considered. For example, the minimum visual angle of the on-screen element on the viewer’s retina must be distinct. Also the concentration of pixels is also an important consideration. Increasing the pixel resolution of a display device (i.e., increasing pixels-per-inch and therefore decreasing the size of the on-screen elements) beyond a certain limit will not produce better result [22].

If the data elements to be shown are large, then a large number of pixels would require to capture the visual features to represent these data elements. The approach that uses in visualization system is to smoothly decreases the number of data elements to be presented as the viewer zooms out and re-displays the elements as the viewer zooms in. Maintaining a balance is required between more data element captured with fewer visual attributes, or fewer data element captured with more visual attributes. Moreover, the intervention of the presence of certain features (e.g., small sizes) with our ability to see other features (e.g., color) also needs to be considered.

Here the questions arise that (i) how many pixels (i.e., what display resolutions) are required for a visual feature to display information effectively, and (ii) what should be the physical size (i.e., what visual acuity) is required to accurately identify and annotate the visual features by our visual system. In order to better understand and validate the appropriate visualization technique, the understanding of the above mentioned limitation are must. The understanding will also characterize to what extent the techniques saturates “visual bandwidth”.

In visualization design, certain characteristics of the dataset and the visual features used to represent its data element needs to be considered:

1. **Dimensionality:** Visual features must be identified to represent the data attributes. For large number of attributes, this may be difficult or impossible to capture each attribute, compelling the display of only a subset of the dataset.

2. **Number of elements:** With the increase of the number of elements to display, it may be impossible to fit them on the screen.

3. **Visual-feature salience:** An effective visualization takes care of the strengths and limitation of visual features to match with suitable data attributes and analysis task to be shown.

4. **Visual interference:** The interference and interaction of different visual features among each other must be eliminated or controlled to guarantee the effective analysis of
data.

The visualization of dataset also gets effected by display resolution and visual acuity, for example, how many data attributes and data element can be represented at once, and how to best display different attributed using suitable visual features.

3.3.1 Display Device Characteristics

Display device characteristics e.g. physical size, viewing distance, display resolution, have a significant effect on the visualization of dataset. This arises two basic question (1) What fraction of dataset can be viewed by the display at any instance? And (2) what fraction of a display can be under consideration of the viewer at any given time? The viewing distance and physical size of the display also affects the visual angle formed by the object which increases with the size of object and decreases with the increase in distance from the eye.

1. Display Resolution  The term resolution has been briefly explained in the beaning of this chapter. This section illustrates impact of the display resolution on visualization. As it has discussed that the amount of information is limited by low display resolution of devices because of the lack of large number of pixels. Just by increasing pixels-per-inch doesn’t provide the required result in terms of the amount of additional information a viewer can see. In contrast, a large field of view (FOV) provided by the large display devices such a powerwall are also under utilize. There is a limitation of the human eye to perceive the amount of information based on horizontal and vertical FOV.

2. Physical Size  Physical size of the device place a significant role to sensory and judgment processes in humans. Available FOV is directly affected by the physical size of the display devices. Large display devices having fixed pixels per inch can display more information because of high resolution .

3. Viewing Distance  As the viewing distance increases, the FOV decreases. FOV is reduced to 30 if the viewing distance is increased by 30-inches.

3.4 Visual Features

Visualization uses variety of visual features to illustrate different data elements. Some of the the most common visual features are hue, texture, luminance, and motion. The following section discusses about these visual properties and their use in different domain, visual interference, and spatial frequency. Figure 3.2 illustrates the use of visual features to represent multidimensional data with an example of weather forecast [23].

3.4.1 Hue

The most commonly used feature in visualization is Color. Color can be described in terms of hue, luminance and saturation. Hue is of the main properties of color, it is explained as the wavelength of the light of the given color. Hue is also explained as “the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, green, blue and yellow.” Saturation is measure of degree of purity or strongness of a color , i.e. how distinguishable the color is from Grey in the HSL and HSV color spaces. The most saturated color is acquired by using just single
Figure 3.1: Visualization of a weather dataset using perceptual texture elements with temperature hue, wind speed density, pressure size, precipitation orientation, and cloud coverage luminance [23]

wavelength at a high intensity while the saturation drops with the drop of intensity of color. Brightness of an image is a function of luminance of a visual target. Luminance is a measure of amount of light emitted from or passes through a particular area. All these dimensions of color needs to be taken care for better control of perceiving difference in various color patterns.

An efficient and balanced color model with linear mapping describes data changes and variations of an attribute’s domain. It is also possible to capture the attribute’s continuous or discrete nature, and spatial frequency using automatic colormap selection algorithm. The color distant, color category, and linear separation must be taken care of while selecting discrete collections of distinguishable colors [23].

Research shows that the observance in the color variation depends on the size and saturation of the color patch along with the contrasting degree from its surrounding colors. Hue is efficient to represent low spatial frequency data of nominal size.

3.4.2 Luminance

Luminance is a physical measure of the brightness of any point on a surface that is radiating or reflecting light. It is often used to define the amount of emission or reflection from flat, diffuse surface. Luminance is best suited in representing high spatial frequency data (i.e. data with high variations in its spatial values). Researchers imposes more importance to luminance over hue in low-level visual system in identifying the boundary of the data. It is observed that our low-level visual system recognize luminance patterns faster than hue patterns.

3.4.3 Texture

Texture is characterize as the appearance, feel or consistency of any surface having a tactile features. Texture is a composition of various perceptual properties. Various display characteristics i.e. density, regularity, contrast, size, shape, orientation, and
direction use to display information. Texture dimension can be controlled using the individual values of data attribute which changes the texture patterns according to the data values.

Height is one of the important characteristics of a texture pattern. Research on cognitive vision shows that the low-level visual system preattentively observe the difference in height. Results proposes the use of this feature to represent the quantitative date and can efficiently support the discrete five values.

Density is another visual parameter for the classification and segmentation of texture. For the representation of low spatial frequency ordinal data, density considers to be best suited. It is observed that hue, luminance and height can interfere with the density of the visual.

The visual system is able to make orientation distinction by using different categories of perceptual direction to capture the orientation. Different researchers have listed different number of categories exists for orientation. For example flat, tiled, and upright or steep, flat, left, and right, then there is 2D orientation which can be used to represent information. Experimental results show that hue and luminance cause visual interference with orientation.

The uniformity of a spatial position of texture element termed as Regularity. It is one of the crucial visual feature that helps in many computer algorithm in performing texture segmentation and classification. The differences in regularity is difficult to detect for a human visual system. Regularity is efficiently represents low spatial frequency data. Hue, luminance, height, and density all these visual features cause visual obstruction for regularity.

3.4.4 Motion
Motion is also a significant feature that holds strong perceptual information. Motion helps in the process of clubbing data and is efficient in providing the overview of pattern and trends in the data. The human visual system is also able to recognize, perceive, track and anticipate movement. Motion is a combination of different other features e.g. Frequency, amplitude, motion shape, direction, phase and velocity.

This chapter presents and discusses the various available small display devices and establish a backdrop for the evaluation of visualization techniques [20].

3.5 Definitions
Different Information visualization techniques can be effectively evaluated in the context of small display devices, when one understands what is meant by small displays. The screen resolution is the parameter to measure the size of the display. Resolution can be explained as the number of pixels (color points) accumulated on a display element. It is the number of pixels on the horizontal axis and vertical axis e.g. the resolution of hand-held device is 96 × 65 pixels.

Sharpness of the image also depends upon the resolution and the physical dimension of the display element. The image will look sharper on smaller display while gradually decrease its sharpness on larger screen for the same resolution because of the spreading
out of the same amount of pixels over a larger space.

The standard CRT desktop displays presents resolution upwards to $1600 \times 1200$ pixels and more. While a typical hand-held devices including PDA (Personal digital Assistant), smartphones usually adopt LCD (liquid crystal display) technology. The resolution of display of these handheld devices ranges typically from $96 \times 65$ pixels to $640 \times 240$ pixels.

When considering the effective size of a display, another parameter i.e. aspect ratio should also be considered along with the resolution. The individual pieces of information presents on the display based on the measure of display resolution, whereas aspect ratio indicates the shape of the display.

The effect of aspect ratio will be explained further below considering a real world example. A street map application running on $75 \times 130$ pixels resolution display will work better than $10 \times 1000$ pixels resolution display, although the amount of pixels in both the display are same. In reality the situation is much more complex, where the types of data being visualized have also given a significant consideration in deciding for the optimal shape of the display. There are other additional factors that contributes to effective display size including the capability of the display element to reproduce colors.

### 3.6 Limitations of Small Display Devices

Adaptation of visualization techniques on small screen devices directly from desktop computers is impossible due to fundamental limitation of these devices. Additionally, the mobility context is different from conventional visualization in many ways. The illustrated below section lists the difference that have to be taken care of in developing visualization application for small display devices [20][21]:

- Limitation in the size, resolution colours of the display.
- The Width/Height ratio is very different from the usual 4:3.
- Limited Computational power, limited hardware (CPU, memory, buses, graphic hardware)
- Human interaction techniques (e.g. tiny keypads, micro-joysticks, rollers) are often inadequate for complex tasks
- The input techniques are different, e.g. hand-writing and pattern recognition on a small surface, one-hand thumb-based input, point-and-tap with stylus
- Bandwidth and Connectivity issues affecting the interactivity of application significantly for large data.
- There is a lack of powerful, high-level graphics libraries.

### 3.7 Physical Characteristics of Display Devices

This section captures the different physical characteristics of display device and visual abilities of the viewer to best match with the visualization design [20].
Different Data attributes could be accurately analyzed and explored using various visual features in the form of images and graphs. Visualization takes advantage of the human low-level visual system to perceive the provided knowledge. In order to take the full advantage of the visual bandwidth of the human visual system, certain criteria needs to be considered:

1. Display device’s physical characteristics (e.g., resolution in terms of the physical size of the display, and the total number of pixels).

2. Acuity of the human visual system (e.g., the distinguish ability limitation of the human eye for different visual features like size and orientation, color, the visual angle on the viewer’s eye subtended by elements).

3. Visualization technique (e.g., the approach used to map a data elements values to a visual representation).

4. Nature of the data (e.g., its dimensionality and number of elements) and the analysis technique to be performed by the viewer.

A lot of research has been done on the visualization techniques and properties of data but very few work has been done on the first two criteria, i.e., how visualization can be enhanced by understanding visual features and visual acuity and how they effect visualization. The foundation for these studies can be provides by the Knowledge from human psychophysics and computer vision [22].

The different visualization algorithm developed to date assumes that the display resolution is sufficient to produce visualization effectively. Rapid increase in technology of small display devices has increased the necessity to select the appropriate visualization algorithm for these devices too. Display devices characteristics, can have a considerable effect on a particular visualization technique. These characteristics i.e physical size, standard viewing distance, pixel resolution, varies across different display devices.

Human visual system is another factor that also needs to be considered. For example, the minimum visual angle of the on-screen element on the viewer’s retina must be distinguishable. Also the concentration of pixels is also an important consideration. Increasing the pixel resolution of a display device (i.e., increasing pixels-per-inch and therefore decreasing the size of the on-screen elements) beyond a certain limit will not produce better result [22].

If the data elements to be shown are large, then a large number of pixels would require to capture the visual features to represent these data elements. The approach that uses in visualization system is to smoothly decreases the number of data elements to be presented as the viewer zooms out and re-displays the elements as the viewer zooms in. Maintaining a balance is required between more data element captured with fewer visual attributes, or fewer data element captured with more visual attributes. Moreover, the intervention of the presence of certain features (e.g., small sizes) with our ability to see other features (e.g., color) also needs to be considered.

Here the questions arise that (i) how many pixels (i.e., what display resolutions) are required for a visual feature to display information effectively, and (ii) what should be the physical size (i.e., what visual acuity) is required to accurately identify and annotate the visual features by our visual system. In order to better understand and validate the
appropriate visualization technique, the understanding of the above mentioned limitation are must. The understanding will also characterize to what extent the techniques saturates “visual bandwidth”.

In visualization design, certain characteristics of the dataset and the visual features used to represent its data element needs to be considered:

1. **Dimensionality:** Visual features must be identified to represent the data attributes. For large number of attributes, this may be difficult or impossible to capture each attribute, compelling the display of only a subset of the dataset.

2. **Number of elements:** With the increase of the number of elements to display, it may be impossible to fit them on the screen.

3. **Visual-feature salience:** An effective visualization takes care of the strengths and limitation of visual features to match with suitable data attributes and analysis task to be shown.

4. **Visual interference:** The interference and interaction of different visual features among each other must be eliminated or controlled to guarantee the effective analysis of data.

The visualization of dataset also gets effected by display resolution and visual acuity, for example, how many data attributes and data element can be represented at once, and how to best display different attributes using suitable visual features.

### 3.7.1 Display Device Characteristics

Display device characteristics e.g. display resolution, physical size and viewing distant, have a significant effect on the visualization of dataset. This arises two basic question (1) What fraction of dataset can be viewed by the display at any instance? And (2) what fraction of a display can be under consideration of the viewer at any given time? The viewing distance and physical size of the display also affects the visual angle formed by the object which is proportional to the size of the object and inversely proportional to the distance of the object from the eye.

1. **Display Resolution** The term resolution has been briefly explained in the beaning of this chapter. This section illustrates impact of the display resolution on visualization. As it has discussed that the low display resolution of devices like mobile phones and PDAs limits the amount of information they can display at any given time because of the availability of less number of pixels. Just by increasing pixels-per-inch doesn’t provide the required result in terms of the amount of additional information a viewer can see. In contrast, a large field of view (FOV) provided by the large display devices such a powerwall are also under utilize. There is a limitation of the human eye to perceive the amount of information based on horizontal and vertical FOV.

2. **Physical Size** Physical size of the device place a significant role to sensory and judgment processes in humans. The physical size of a display device has a direct affect on the available FOV. Also, for a fixed pixels-per-inch, larger display devices have higher resolutions and therefore may be capable of visualizing more information.
3. Viewing Distance  As the viewing distance increases, the FOV decreases. For example, a 16-inch display placed 22-inches from the user produces a FOV of approximately 40°. Increasing the viewing distance to 30-inches reduces the FOV to 30°.

3.8 Visual Features

Visualization uses variety of visual features to illustrate different data elements. Some of the the most common visual features are hue, texture, luminance, and motion. The following section discusses about these visual properties and their use in different domain, visual interference, and spatial frequency. Figure 3.2 illustrates the use of visual features to represent multidimensional data with an example of weather forecast [23].

![Figure 3.2: Visualization of a weather dataset using perceptual texture elements with temperature hue, wind speed density, pressure size, precipitation orientation, and cloud coverage luminance [23]](image)

3.8.1 Hue

The most commonly used feature in visualization is Color. Color can be described in terms of hue, luminance and saturation. Hue is of the main properties of color, it is explained as the wavelength of the light of the given color. Hue is also explained as “the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, green, blue and yellow.” Saturation is measure of degree of purity or strongness of a color, i.e. how distinguishable the color is from Grey in the HSL and HSV color spaces. The most saturated color is acquired by using just single wavelength at a high intensity while the saturation drops with the drop of intensity of color. Brightness of an image is a function of luminance of a visual target. Luminance is a measure of amount of light emitted from or passes through a particular area. All these dimensions of color needs to be taken care for better control of perceiving difference in various color patterns.

An efficient and balanced color model with linear mapping describes data changes and variations of an attribute’s domain. It is also possible to capture the attribute’s
continuous or discrete nature, and spatial frequency using automatic colormap selection algorithm. The color distant, color category, and linear separation must be taken care of while selecting discrete collections of distinguishable colors [23]. Research shows that the observance in the color variation depends on the size and saturation of the color patch along with the contrasting degree from its surrounding colors. Hue is efficient to represent low spatial frequency data of nominal size.

3.8.2 Luminance

Luminance is a physical measure of the brightness of any point on a surface that is radiating or reflecting light. It is often used to define the amount of emission or reflection from flat, diffuse surface. Luminance is best suited in representing high spatial frequency data (i.e. data with high variations in its spatial values). Researchers impose more importance to luminance over hue in low-level visual system in identifying the boundary of the data. It is observed that our low-level visual system recognize luminance patterns faster than hue patterns.

3.8.3 Texture

Texture is characterize as the appearance, feel or consistency of any surface having a tactile features. Texture is a composition of various perceptual properties. Various display characteristics i.e. density, regularity, contrast, size, shape, orientation, and direction use to display information. Texture dimension can be controlled using the individual values of data attribute which changes the texture patterns according to the data values.

Height is one of the important characteristics of a texture pattern. Research on cognitive vision shows that the low-level visual system preattentively observe the difference in height. Results proposes the use of this feature to represent the quantitative date and can efficiently support the discrete five values.

Density is another visual parameter for the classification and segmentation of texture. For the representation of low spatial frequency ordinal data, density considers to be best suited. It is observed that hue, luminance and height can interfere with the density of the visual.

The visual system is able to make orientation distinction by using different categories of perceptual direction to capture the orientation. Different researchers have listed different number of categories exists for orientation. For example flat, tiled, and upright or steep, flat, left, and right, then there is 2D orientation which can be used to represent information. Experimental results show that hue and luminance cause visual interference with orientation.

The uniformity of a spatial position of texture element termed as Regularity. It is one of the crucial visual feature that helps in many computer algorithm in performing texture segmentation and classification. The differences in regularity is difficult to detect for a human visual system. Regularity is efficiently represents low spatial frequency data. Hue, luminance, height, and density all these visual features cause visual obstruction for regularity.
3.8.4 Motion

Motion is also a significant feature that holds strong perceptual information. Motion helps in the process of clubbing data and is efficient in providing the overview of pattern and trends in the data. The human visual system is also able to recognize, perceive, track and anticipate movement. Motion is a combination of different other features e.g. Frequency, amplitude, motion shape, direction, phase and velocity.
Chapter 4

Comparative Analysis of Information Visualisation Method on Tablet PC

In chapter II, a detailed study is presented on the working and interaction mechanism of seven different data visualisation techniques. In this chapter, a comparison study is presented on four most common visualisation layout: (i) Space Tree, (ii) Hyperbolic Tree, (iii) Icicle Plot, and (iv) Treemap. Two of the four design is chosen from node-link technique while the other two is a space-filling visualisation technique. The remaining discussed visualisation technique is the modification or alteration of the chosen four methods.

Following section briefly captures the use of these visualisation techniques in various application. It also discusses the evaluation and comparison work that has been done by different researchers on various other layouts on different devices.

4.1 Related Work

There is little literature available, which discusses about the use of Treemap and Hyperbolic Tree technique for small screen devices. Khella et al., employed the Treemap visualization technique in developing a pocket pc image browser ‘Pocket PhotoMesa’. The approach used Quantum strip Treemaps for rendering images with zoomable user interface for navigation on PDAs [25]. Engdahl et al. used squarified Treemaps to visualise the threads in discussion forums, as colored rectangle on PDAs, to effectively utilize limited space. He compared Treemap versions with traditional text-based tree structure. Results preferred Treemap over traditional way, as the user is easily able to grasp the content of the discussion forum, using treemap [26]. Keranen et al. [27] also used Treemap layout for displaying data on small screen device. The author proposed an adaptive method to divide the screen size at runtime, by considering the device display size and currently active component of user interface. citejob used the Hyperbolic Tree structure to evaluate the multi touch of Microsoft Windows 7 platform. The approach presented in the paper, ported the existing Software Product Line Engineering (SPLE) and Variability Management application in Microsoft Surface platform, to Microsoft Windows 7 platform [28].

Few efforts have been made to provide the comparative analysis of these visualization
method on desktop machine. In [29] Barlow et al. compared four 2D-visualization methods of hierarchical structure. The result suggested that Icicle performed better than organisation chart in some task. However, there were no significant differences in user responses observed among Icicle, Ring Tree and Organisation Chart. All these methods are preferred by users over Treemap. Plaisant et al. [30], compared 3 tree-browsing techniques e.g. Space Tree, Microsoft Explorer and Hyperbolic Tree browser. It was observed that users found Space Tree more attractive than Explorer. Space Tree performed better for both topology and navigation tasks, even though both have almost similar interfaces. Andrews et al. [31], reviewed Hyperbolic tree browser, Windows Explorer, Information pyramid browser, and treemap visualization techniques on medium size display with 1024×768 actual display resolution. The techniques were compared on nine different factors e.g. usability, navigation, operability, intuitiveness. Traditional tree browser approach is preferred the most, while Treemap is least preferred. No significant work has been done to analyse the performance and usability of these visualization layouts, on small devices like Tablet PCs.

4.2 Evaluation of visualization layout for Tablet PCs

The aim of this work is to transport the existing visualization methods on Tablet PCs, and analyse their performance on such devices. Experiments were conducted to learn the usability and effectiveness of these layouts in performing information searching tasks.

4.2.1 Experimental Setup

In the experiment, 4 tree-browsing layouts were compared: Hyperbolic tree, Icicle Plot, Space Tree, and Treemap. All the experiments were done on an Datawind UbiSlate 7Ci Tablet PC (Aakash tablet) of 800×480 screen resolution with 1GHz CPU, 512MB of RAM, running Android 4.0.3 [32]. It is a 7-inches touch-screen device with finger touch (or stylus) as a primary means of control. Each visualization occupied 636×296 pixels on display area. The layouts were developed using Javascript Infoviz toolkit (JIT) [33], an open source library for interactive data visualization on web. The application was loaded on user tablet to avoid delays due to network connection issues and provide uniform means for comparison.

4.2.2 Participants

The experiment was conducted with seventeen test users ranging from 20 to 30 years of age. Twelve of the participants were male, and five were female. All the participants were students in different disciplines and had 15 to 20 years of experience of using computers. Nine of the participants were students from computer science, five students were from industrial design, and three of them were bioscience students. None of the participants had any experience with Hyperbolic Tree, Icicle Plot, Space Tree or Treemap visualization.

4.2.3 Design

The experimental set-up consisted of 4(layout)×7(tasks) with different set of data to each layout. The analysis is based on repeated-measure with-in subject design. Each dataset was repeated equal number of times with all four tree layouts to counterbalance the layouts and provide equal basis for comparison. Data was collected from online data repository in four different category: (i) online shopping database, (ii) data on different
spoken languages in the world, (iii) World top universities ranking in last three years, and (iv) Film awards database. Variable structure and size of datasets were used to analyse the performance of each layout. Table 4.1 lists the variations in size and topology of each database.

Table 4.1: TEST DATA CATALOGUE

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size (nodes)</th>
<th>Hierarchical levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online shopping database</td>
<td>130</td>
<td>5</td>
</tr>
<tr>
<td>World spoken languages</td>
<td>750</td>
<td>6</td>
</tr>
<tr>
<td>World top universities ranking</td>
<td>330</td>
<td>7</td>
</tr>
<tr>
<td>Film awards database</td>
<td>810</td>
<td>6</td>
</tr>
</tbody>
</table>

The text of the labels were kept smaller to avoid clustering of nodes. A user could navigate through the tree by tabbing on the node’s label. To increase readability and interactivity, the font size of the label was set at 12 pt. All the datasets were of feasible size to avoid overlapping of the nodes. Automatic adjustment technique was provided to balance the spacing between the nodes at same or different levels of hierarchy. Only upto four levels of tree hierarchy were visible to the user at one time to avoid showing bulk of data. The design features of the four layouts discussed, are listed in Table 4.2.

Table 4.2: DESIGN FEATURES OF LAYOUT

<table>
<thead>
<tr>
<th>Layout</th>
<th>Next Node Event</th>
<th>Previous Node Event</th>
<th>Level Shown (initial)</th>
<th>Level shown (max.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperbolic tree</td>
<td>tab,click (label)</td>
<td>tab,click (label)</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Icicle</td>
<td>tab,click (area)</td>
<td>触碰并保持 (area)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Space Tree</td>
<td>tab,click (label)</td>
<td>tab,click (label)</td>
<td>1</td>
<td>全部</td>
</tr>
<tr>
<td>Treemap</td>
<td>tab,click (area)</td>
<td>触碰并保持 (area)</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Experiments were performed on each user separately, under the supervision of a volunteer. Participant’s behaviour and their interaction with the tools, were observed closely. A break of 20 to 30 minutes was given for each user between experiments.

4.2.4 Tasks
Users were asked to perform seven different tasks for each view listed in Table 4.3. There were no limitation of response time for any of the tasks. Participants were briefed for few minutes about the set-up and anticipation of the experiment. Design and interaction
method of each layout, were also explained before the task. All the tasks were aimed to understand the working and behaviour of each tree view in the following four categories:

**Tree Overview**
User understanding about the topology of the layout was tested with three different tasks.

- **Balanced or Unbalanced Tree:** Participants were asked to find whether the tree was balanced or unbalanced. They were explained that all the leaf nodes in a balanced tree will be at the same level or two consecutive levels, while this is not the condition in an unbalanced tree.

- **Height of the Tree:** Participants were told to find the maximum number of hierarchical levels in the tree.

- **Bushiest Child node:** Test users were asked to select a node with most number of leaf nodes.

**Data Search**
Participants were given two data searching tasks for each layout.

- **Find Parent Node** Given a node, users were asked to find the parent of the node.

- **Depth of a Node** Participants were given a node in the hierarchy, they were asked to find the level at which the node was presented.

**Data Comparison**
Participants were explained different layouts to find largest node. In the case of Treemap and Icicle, the area of the node corresponds to the value of the node, while in hyperbolic tree and Space Tree, the label contains the value of the node.

**Data Counting**
In this task, participants were given a node in the tree and were asked to count the child nodes of the node.

**4.2.5 Result**
Users’ Response Times (RT) for each task were noted in seconds. Table 4.4 lists average task completion time and Standard Deviation (SD) for each layout and task. Fig. 4.1 shows the chart of response time of each view for each task. The normalized response time is plotted by using the log-transformation of users’ response time. Graphs shows that the mean completion time for tasks $T_4$ and $T_5$ are almost similar in Icicle Plot and hyperbolic tree. Performance of Treemap was significantly better among all other layouts for overview and comparison tasks followed by Hyperbolic tree. Icicle method took longer time for searching and comparison tasks e.g. $T_3$ and $T_6$. Response time (RT) of Space Tree was consistent throughout the tasks among all layouts.
Table 4.3: List of tasks in the experiment design

- \( T_1 \): Is tree balanced or unbalanced?
- \( T_2 \): Find the height/depth of the tree.
- \( T_3 \): Find the bushiest node in the deepest level.
- \( T_4 \): Name parent Node of the given node.
- \( T_5 \): Find the given node level in the tree.
- \( T_6 \): Find largest node in the tree.
- \( T_7 \): Count total child nodes of a given node.

Kolmogorov-Smirnov (K-S) is one of the most useful techniques, to analyse data distribution. Table 4.4 demonstrates the distribution of data for repeated-measure analysis of each task. The critical value for the analysis is taken as \( p=0.05 \). K-S values for tasks \( T_4 \) and \( T_5 \) in Space Tree, are below critical value, indicates the non-normal distribution of the observed value. Similar case is with Icicle, Treemap and Hyperbolic tree for task \( T_5 \). K-S value for all other tasks are above the critical value, signifies the normal distribution of data.

Fig. 4.2 displays the percentage correction of users’ responses received for each task for each view. The chart depicts that most of the user gave incorrect responses for comparison task (\( T_6 \)) in Hypertree and Icicle Plot. There was no significant difference in the users’ performance in searching task in all four views. Treemap performance was better than Hyperbolic tree and Icicle Plot in most of the tasks. The users’ performance was lowest for topological tasks in Icicle Plot among all views. Space Tree had consistently higher correctness percentage for all tasks.

Table 4.4: Mean task completion time of each task for each tree layout. The One-Sample Kolmogorov-Smirnov Test for normality check of the data distribution

<table>
<thead>
<tr>
<th>Task</th>
<th>Hyperbolic Mean</th>
<th>K-S Mean</th>
<th>Icicle Mean</th>
<th>K-S Mean</th>
<th>Space Mean</th>
<th>K-S Mean</th>
<th>Tree Map Mean</th>
<th>K-S Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_1 )</td>
<td>28.28</td>
<td>0.200</td>
<td>97.14</td>
<td>0.212</td>
<td>58.3</td>
<td>0.243</td>
<td>23.13</td>
<td>0.301</td>
</tr>
<tr>
<td>( T_2 )</td>
<td>45.7</td>
<td>0.527</td>
<td>86.16</td>
<td>0.565</td>
<td>50.3</td>
<td>0.456</td>
<td>35.5</td>
<td>0.675</td>
</tr>
<tr>
<td>( T_3 )</td>
<td>158.6</td>
<td>0.378</td>
<td>212.14</td>
<td>0.340</td>
<td>88.3</td>
<td>0.243</td>
<td>68.3</td>
<td>0.675</td>
</tr>
<tr>
<td>( T_4 )</td>
<td>34.3</td>
<td>0.215</td>
<td>36.41</td>
<td>0.239</td>
<td>28.3</td>
<td>0.023</td>
<td>58.4</td>
<td>0.243</td>
</tr>
<tr>
<td>( T_5 )</td>
<td>12.2</td>
<td>0.034</td>
<td>10.41</td>
<td>0.043</td>
<td>18.3</td>
<td>0.023</td>
<td>46.4</td>
<td>0.0243</td>
</tr>
<tr>
<td>( T_6 )</td>
<td>102.4</td>
<td>0.521</td>
<td>94.1</td>
<td>0.542</td>
<td>83.4</td>
<td>0.452</td>
<td>130.3</td>
<td>0.453</td>
</tr>
<tr>
<td>( T_7 )</td>
<td>8.4</td>
<td>0.435</td>
<td>10.41</td>
<td>0.443</td>
<td>15.3</td>
<td>0.130</td>
<td>45.5</td>
<td>0.321</td>
</tr>
</tbody>
</table>

Users were also asked to rank each design on seven factors on a 5-point scale. List of seven design factors is presented in Table 4.5 with representation of each scale value on 1
Figure 4.1: User logarithmic Response time in seconds for each task on following four views: Hyperbolic Tree (HT), Icicle Plot (IP), Space Tree (ST), Treemap (TM). Response time includes correct and incorrect response.
to 5 Likert scale. Table 4.6 summarizes the subjective ranking for each layout on seven design factors. The two columns in Table 4.6 are Mean ratings and Standard Deviation of each layout on seven design factors. Treemap was ranked lowest by all the participants as a difficult and non-intuitive design. However, after having familiarity with the layout, they preferred Treemap over other layouts for topological tasks for providing better overview of the data structure. Users were having difficulty in navigation through the hyperbolic tree, as depicted in the Table 4.6. Transformation between the levels of the hyperbolic tree is not smooth, hence users preferred Icicle and Space Tree, on the factor of navigation and orientation. Since the layouts were displaying on a small rectangular area, participants did not prefer hypertree on the usability factor, because of the wastage of display area. Table 4.6 shows that the performance of Space Tree is consistent for all tasks. The statistics shows that users favoured Space Tree over Icicle and Treemap, while Hypertree was liked the least.

Table 4.5: List of preference parameters & Representation of 5 - point Likert scale

<table>
<thead>
<tr>
<th>Factors</th>
<th>5-Point Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$: Overview</td>
<td>1 = strongly disagree</td>
</tr>
<tr>
<td>$F_2$: Understandable</td>
<td>2 = disagree</td>
</tr>
<tr>
<td>$F_3$: Intuitive</td>
<td>3 = undecided</td>
</tr>
<tr>
<td>$F_4$: Usable</td>
<td>4 = agree</td>
</tr>
<tr>
<td>$F_5$: Navigation</td>
<td>5 = strongly agree</td>
</tr>
<tr>
<td>$F_6$: Operable</td>
<td></td>
</tr>
<tr>
<td>$F_7$: Orientation</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: Mean and Standard Deviation of ratings of participants’ preference parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Hyperbolic</th>
<th></th>
<th>Icicle</th>
<th></th>
<th>Space</th>
<th></th>
<th>Tree map</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td>Mean S.D</td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td>3.6 1.2</td>
<td>2.1 1.3</td>
<td>2.9 2.4</td>
<td>4.1 3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understandable</td>
<td>3.3 2.4</td>
<td>3.8 1.1</td>
<td>4.5 2.2</td>
<td>2.6 1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intuitive</td>
<td>3.6 4.0</td>
<td>1.0 4.9</td>
<td>2.0 2.0</td>
<td>1.8 2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usable</td>
<td>1.6 2.1</td>
<td>3.4 2.3</td>
<td>4.8 2.5</td>
<td>2.6 1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation</td>
<td>1.4 1.2</td>
<td>2.5 2.2</td>
<td>4.2 2.2</td>
<td>2.4 1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operable</td>
<td>2.8 1.2</td>
<td>3.3 2.1</td>
<td>4.7 1.7</td>
<td>1.8 1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orientation</td>
<td>1.4 1.2</td>
<td>4.1 2.8</td>
<td>4.8 2.2</td>
<td>3.7 1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Discussion

4.3.1 Design

The expected results were considered to be in favour of Treemap, before conducting experiments. Initially our analysis was considering Treemap to effectively utilise the small display area. However, a significant difference was observed in Participants’ response in using Treemap. Users did not favour the idea of showing complete data at once on a small screen. It was observed that users were exposed to huge amount of unwanted data, causing distraction for them. Navigation through the tree nodes was also difficult due to clustering and overlapping of the labels due to small node size. Users preferred Icicle Plot over Treemap, for better utilisation of available display space. However, Treemap was chosen as a better design for comparative tasks, as it shows complete hierarchy of the tree. Space Tree with collapsible nodes was highly preferred for small screen device, as it allowed user to see only the relevant data. Hyperbolic tree was not considered suitable for devices with screen aspect ratio 16:9 or 16:10, because of its circular structure.

4.3.2 Interactions

Several important conclusions were drawn from the participants’ interactions with the system. It was noted that the user faced difficulties in interacting with the layouts, based on space-filling technique. It was observed that in Treemap and Icicle Plot, many times nodes were unintentionally clicked by the users’ thumb or palm touch. One possible solution to this problem is to only enable the clicking to a specific area of the node e.g. label of the node. Touch enabled devices also have limited available touch events that could be used for tree navigation. Many touch events are already in use by the devices for user interactivity e.g. double-click, drag, touch-move. Single-click and touch-hold were the interaction events, used to access child nodes and parent nodes respectively in the tree.
the Icicle Plot and Treemap layout. From users’ responses, it was observed that users
find the touch-hold event as annoying and time consuming.
Chapter 5

WebVis - An Interactive Web-based Data Visualisation Tool

This chapter demonstrates the design and implementation of a Web based interactive data visualisation tool. The idea is inspired from a public Web site Many Eyes, to create interactive visualisation, developed by IBM Research and IBM Cognos software group [34]. It provides users a platform to upload and share data. Each visualisation is provided with a discussion forum to foster a social style of data analysis.

5.1 An overview of the WebVis site

WebVis, an interactive Web based data visualisation tool, is designed with roughly similar model as well-known participatory sites such as Many eyes [34], Youtube [35], and Flicker [36]. Data visualisation and exploration is the heart of the tool. Data uploading, constructing visualisation, sharing of data, interacting and analysing the data are central activities on the site. Each of these activities are explained in details in subsequent sections.

WebVis, like above discussed Web standard based tool, provides user to view their data in interactive visualisation from externally uploaded file. In addition to this, WebVis also facilitates users to query and explore data from their own database. Designing query on the fly aids in retrieving only the required data from the bulk of information. Hence, it enhances more understanding of the data and decreases time for searching and navigating through huge data.

To navigate through the different portions of Website, an interactive slider view is used as shown in Figure 5.1, that provides three different mechanism for uploading and viewing data (i) Upload and View data using external file (ii) Design dynamic query to Database, (iii) View data by stored database query.

5.2 The Data Model

The core data model used by WebVis, is hierarchical data: that is a set of named column with a hierarchical relationship in the data of each column. Such system offers potential benefit to the users where user can upload and view their own data. The benefit ranges
from individuals to a large number of groups. The tool is useful in various application and can serve different purposes. For instance, teacher might use the visualisation to see the scores of each student on Treemap and can identify the highest scorer. It might help students to search for books on a particular topic in a particular subject. An already made query, designed by teacher might help such students in finding the relevant data. A platform of shared information encourages participation of students and create a healthy and interactive learning environment. It serves as a medium for students as well as teacher to know what type of information is mostly required.

In the tool, the focus is on representing two modes of hierarchical data: one is textual relational data e.g. book catalogue, course outline etc. and other is relational numeric data e.g. students’ marks in a course, course registration data etc. A balance between complex and conflicting constraints on the design needs to be taken care by the tool to achieve these benefits. For example, in the later type of data, a specific data format is required by the corresponding suitable visualisation tool for such types of data e.g. Treemap, Icicle etc. User is entitled to provide a numerical data column named as area representing the frequency, score, magnitude etc. Care has been taken in providing an easy and understandable data model format to the end-user. Knowing the fact that the data is uploaded and database is queried by end-user, the interface is designed such that is intuitive and understandable for a non-programmers as well. At the same time, data needs to be manipulated to be flexible enough to express the data structure required by each visualisation such as Spacce tree, Treemap algorithm.

5.3 Data Uploading

Users can directly upload data from an external file on the web site as shown in Figure 5.4. Currently the tool supports only the excel file format. The file’s tabular structure as shown in Figure 5.2, is then converted to JSON (JavaScript Object Notation) structure as presented in Figure 5.3, which feeds to the visualisation layout as input. Each column in the tabular structure is mapped to the corresponding level of the tree. Whereas, the distinct values in each column, mapped to the node in the visualisation tree. First column of the tabular file, is mapped to the first level of the hierarchy and so on. Root node is considered at level 0.

As discussed in the last section, a specific format of the uploaded data, is required by the tool. Users’ data should include an area field containing the numeric data as the last column. The tool includes the area column in the uploaded with a constant numeric value while constructing the JSON structure. Hence, the system continues to work and the information it might convey is the number of sub-nodes at any level.

An other method to visualise user data is by storing it in the database and upload only the required information from the database by dynamically queries created by users from front-end. A detailed demonstration on creation of the dynamic query from the
5.4 Working with Data

Each data file that is uploaded on the WebVis site, is resided on the server for future use. The list of uploaded file is shared among all users and also allows user to update the content of the file, thereby encourages collaboration among users. However, a mechanism for imposing security rights can be incorporated within the system to restrict user access with the data.

5.5 Visual Mapping

A visualization is created by matching a dataset with a visualization component. Hence different visualisation is capable of displaying different types of dataset. A Space tree, for example is not feasible for viewing numeric data while suited to display number of textual columns as multiple levels of a tree structure. Similarly, a treemap requires a number of textual columns to define its hierarchy and a numerical columns that map to size, whereas different colors in treemap represents different level of hierarchy and helps in navigation through the tree.

5.6 Designing Database Query

Figure 5.5 shows the interface for the users to design database query from the front-end. The treemap view displays list of all database tables, along with list of columns under each tables, in the chosen database. User might select any number of columns from any table or combination of different column from multiple tables. User can also filter the data further by using additional searching conditions.

A user interface for advance search is also included in the tool as presented in Figure 5.5. User can query any specific data or a range of data by selecting the appropriate
```json
{
  "id": "node02",
  "name": "Shopping",
  "data": {
    "$area": 8,
    "$dim": 8,
    "$color": "#001eff"
  },
  "children": [
    {
      "id": "node13",
      "name": "Books",
      "data": {
        "$area": 65,
        "$dim": 65,
        "$color": "#9554ff"
      },
      "children": [
        {
          "id": "node24",
          "name": "Business Administration",
          "data": {
            "$area": 30,
            "$dim": 30,
            "$color": "#ee6aff"
          }
        }
      ]
    }
  ]
}
```

Figure 5.3: JSON file format for Hierarchical Data

Figure 5.4: A View of External File Uploading Interface
matching condition. All possible matching conditions e.g. *equals to, less than or equal to, greater than or equal to, and not equal to*, are also provided in the query designing interface. Each condition is given with a text box for the user to input match condition(s) for the data. The implementation of the query design interface is explained in the following sub-section.

![Figure 5.5: A View of User Interface for Accessing Database with Query Generation and Advance Search Functionality](image)

### 5.6.1 Implementation of Query Design Interface

*MySQL* is used as a back-end database for storing users’ data. *Information_schema* is the information database in *MySQL* and provides access to database metadata. List of tables with nested lists of columns on the *Query design* user interface, is displayed from the *Information_schema* database. The code snippet with the explanation of working of the query, is given in Table 5.6.1.

When the user clicks on *Show Data* link, all the user’s selected column with additional data filtering conditions, if provided, are sent to *MySQL* to fetch user queried data. Table 5.2 contains the code snippet for the *MySQL* stored procedure for fetching data required by user.
Table 5.1: Code snippet: User Query

```sql
CREATE TEMPORARY TABLE mytemp(tbl VARCHAR(255),tblcol VARCHAR(255),reftbl VARCHAR(255),reftblcol VARCHAR(255)) ;

-- index starts from 1 e.g substr(str, startIndex, len)
SET strTablelist=SUBSTRING(Param1,1,INSTR(Param1,'SELECT')-1);

SET strColumn = SUBSTRING(Param1,INSTR(Param1,'SELECT'),strwhr);

SET strCondition=CASE INSTR(Param1,'WHERE') WHEN 0 THEN '' ELSE SUBSTRING(Param1,INSTR(Param1,'WHERE'),LENGTH(Param1)-INSTR(Param1,'WHERE')+1) END;

SET @s=CONCAT('INSERT INTO mytemp(tbl,tblcol,reftbl,reftblcol)
SELECT k.TABLE_NAME,k.COLUMN_NAME,k.REFERENCED_TABLE_NAME,k.REFERENCED_COLUMN_NAME
FROM informinipagemation
WHERE k.TABLE_NAME IN (' ,strTablelist, ' )');
```

Initially, the query parse the input parameters of the MySQL stored procedure, and list all the table’s name in a temporary MySQL table. Stored Procedure, then uses KEY_COLUMN_USAGE table from Information_schema database, to find out the list of any REFERENCED_TABLE and REFERENCED_COLUMN to resolve the FOREIGN_KEY_CONSTRAINT, if any.

Table 5.2: Code snippet: MySQL Procedure for Fetching User Queried Data

```sql
Block2: BEGIN
  DECLARE FindFKtable_cur CURSOR FOR SELECT * FROM mytemp;
  DECLARE CONTINUE HANDLER FOR NOT FOUND
  BEGIN
    SET no_more_rows = TRUE;
    OPEN FindFKtable_cur;
    select FOUND_ROWS() into num_rows;
    the_loop: LOOP
      FETCH FindFKtable_cur INTO table_val, column_val, reftable_val, refcolumn_val;
      IF no_more_rows THEN
        CLOSE FindFKtable_cur;
        LEAVE the_loop;
      END IF;

      -- Check if the table_val is a reference table for any other table
      SET Istbref=(SELECT COUNT(*) FROM mytemp where reftbl=table_val);
  END BEGIN;
```

51
IF(LENGTH(strTableJoin) = 0) THEN
    IF (reftable_val IS NOT NULL) THEN
        SET strTableJoin = CONCAT(table_val, ' INNER JOIN ',
                           reftable_val, ' ON ' . table_val . '.' . column_val . '=' .
                           reftable_val . '.' . refcolumn_val);
        SELECT strTableJoin;
    ELSEIF (reftable_val IS NULL) THEN
        SET strTableJoin = table_val;
    END IF;
ELSEIF(length(strTableJoin) <> 0) THEN
    IF(instr(strTableJoin.table_val)=0 AND Istblref=0) THEN
        IF (reftable_val IS NULL) THEN
            SET strTableJoin = CONCAT(strTableJoin, ',
                           table_val);
        ELSEIF(reftable_val IS NOT NULL AND instr(strTableJoin, reftable_val)=0) THEN
            SET strTableJoin = CONCAT(strTableJoin, ',
                           table_val, ' INNER JOIN ', reftable_val, ' ON '
                           . table_val . '.' . column_val . '=' . reftable_val . '.'
                           . refcolumn_val);
        ELSEIF(reftable_val IS NOT NULL AND instr(strTableJoin, reftable_val)<>0) THEN
            SET strTableJoin = CONCAT(strTableJoin, ', INNER
                           JOIN ', table_val, ' ON ', table_val . '.' . column_val . '=' . reftable_val . '
                           . refcolumn_val);
        END IF;
    ELSEIF(instr(strTableJoin.table_val)<>0) THEN
        IF(reftable_val IS NOT NULL AND instr(strTableJoin . refcolumn_val)=0) THEN
            SET strTableJoin = CONCAT(strTableJoin, ', INNER
                           JOIN ', table_val, ' ON ', table_val . ',' . column_val . '=' . reftable_val . '
                           . refcolumn_val);
        ELSEIF(reftable_val IS NOT NULL AND instr(strTableJoin . reftable_val)<>0) THEN
            SET strTableJoin = CONCAT(strTableJoin, ',
                           table_val .', column_val . '=' . reftable_val . ',
                           refcolumn_val);
        END IF;
    END IF;
END IF;
END IF;

        -- count the number of times looped
    SET loop_cntr = loop_cntr + 1;
    IF (loop_cntr > num_rows) THEN set no_more_rows = TRUE; END IF;
END LOOP the_loop;

END Block2;
The dataset fetched from the above discussed procedure in Table 5.2 is sent back to be converted into JSON data format which finally feed into the user’s chosen visualisation layout. Figure 5.3 shows the code snippet for returning user queried dataset.

Table 5.3: Code snippet: Returning User Queried Dataset

```sql
SET strQuery=CONCAT(strColumn,' FROM ',strTableJoin,IFNULL(strCondition,''));
SET @WholeQuery = strQuery;
PREPARE stmt2 FROM @WholeQuery;
EXECUTE stmt2;
```

5.7 Features of Visualisation Layout for User Interaction

Several new features are added in the standard JIT - JavaScript Infovis Toolkit library by observing users’ requirements while conducting experiments discussed in previous chapter. User is provided with options to view either portions of the tree or full view by using Collapse/Full View feature, explained in subsequent sections. There are few other useful features added to enhance the users’ interactivity with the system and provide an easy access to the required data. Usefulness of the system can only be increased by employing new and innovative method of extracting users’ required data. In the following sub-section additional incorporated features in the layouts are discussed.

5.7.1 Tracking the Node Path to Root

In the Space tree, a feature is provided to track the path of the currently opened node from the root node as highlighted in Figure 5.6. This feature helps users to track the location of the node with respect to the root node and provides a view of the ancestors of the currently selected node.

5.7.2 Search a Node in the Tree

A node finding technique is provided in the layout to quickly locate a user given node. Figure 5.6 shows the user searched node in the tree.

5.7.3 Collapse/Full View

User is provided with a customisable view for each layouts. A technique to give user collapsible or a full tree view is incorporated specific to each layouts.

**Space Tree**  In this layout, users can choose among three different views of the layout, suitable to their needs. *Full View* renders complete tree structure on page load. This type of view is appropriate for smaller tree structure with labels of smaller length. In the *Navigate View*, users are only provided with top levels’ structure of the tree and further tree’s data is opened only on-demand. In this design, the branches will remain open if once opened. It helps in data comparison task among nodes in multiple branches. Whereas in *Collapse View*, the other branches will collapse on the opening of other
branch. This layout is feasible for large tree hierarchy. It helps in data searching and navigation tasks. Figure 5.6 shows controls to see different views of Space Tree.

**Treemap** Three different views of treemap layout are developed in the design to achieve following users’ requirement. In *Full View* represented in Figure 5.7, complete tree structure with nodes’ labels is visible to the user on page load. This is suited for smaller depth hierarchical tree with smaller size labels. The *Tree View* displays completes tree structure at once, on the page load and provides broader view of tree structure as shown in Figure 5.8. However, labels are displayed only up to few levels. This technique helps in displaying tree structures with large size nodes’ labels. It prevents data clustering by hiding labels of deeper levels’ node. For large tree structure with longer nodes’ label, *Collapsible View* is suitable explained in Figure 5.9. It only displays few levels of tree hierarchy with their nodes’ labels. User can expand or collapse further tree’s level on need.

### 5.7.4 Pop-up Window for Displaying Audio/Video

The tool also embeds functionality to access detail content of the node while continue exploring with the data. If the node contains link to an external web-page or path to an audio/video file, the tool opens these file in a pop-up window as presented in Figure 5.10.
Figure 5.7: Treemap: Full View of Tree Structure

Figure 5.8: Treemap: Collapsible View of Tree Structure

Figure 5.9: Treemap: Collapsible View of Tree Structure without Nodes’ Details
Figure 5.10: Pop-up Window for Displaying Files/Web-Page

Overview of the Syllabus for Computer Networks

- Motivation: goals of networking, well-known applications such as web, e-mail and ftp => need for a layered architecture, OSI and Internet.
- Host-to-host communication: RS-232 over serial line; handshaking and error handling; packet switching; reliable transmission - stop-and-wait, sliding window; logical connections.
Chapter 6

Summary

6.1 Summary

In the recent passed, several new and innovative visualisation techniques have emerged for displaying large hierarchical data. Few of them were meant to show large volume of data in a limited display size termed as Space-filling technique. Others were efficient in providing traditional tree layout in a more interactive way, known as Node-link technique. These data visualisation techniques have been used in many desktop application and their performance have also been evaluated for medium to large size display devices. Very few mobile based applications have also employed some of these visualisation layout e.g. treemap. Rapid increase in the usage of Tablet PC devices in the recent years, necessitates to learn the applicability of these visualisation techniques on these devices. Hence, an effort is made in this project to deploy the existing data visualisation method on Tablet PC and learn their usability on such devices.

Few experiments were conducted to evaluate these visualisation methods and technologies used to design these layouts as discussed in previous chapters. However, the actual statistics of the usefulness and efficiency of such techniques can be known only if the system is being used in the real scenario. For this purpose, a data visualisation tool WebVis, is designed based on web standards. It provides a platform for the user, to view and interact with their data. Various features have been incorporated in the tool for uploading and customising view of data. With the use of well structured data, this tool is capable of exploring hidden patterns in the data by presenting its visual aspect. It assists users to better understand the topology and structure of the data. It provides more interactivity in searching and data navigation task

Among various limitation of small screen devices, limited memory is also a critical factor. It limits the amount of data to be displayed and also effects user interactivity and layouts’ animation. Hence, user needs to take care of amount of data to be fetched. For this purpose, a data filter technique is also employed in the tool. In this way, user is only needs to search required data from large volume of data and also devices’ interactivity improves. However, care needs to be taken while designing query for data fetching. An incompatible data field selection may lead to meaningless data structure.
7.1 Conclusions

Data representation and analysis is benefited by numerous advancements in the visualization field. These innovative visualization techniques have been applied to many different applications from many decades. The motivation in the above presented work was to transport the existing visualization technique to Tablet PCs, and examine their usability and performance on these devices. Experiments were conducted to analyse four common tree visualization techniques: Hyperbolic tree, Icicle, Space Tree, and Treemap. The layouts were compared on the basis of practicability and feasibility on the following factors: overview, searching, comparison, and user preferences. Results suggested that the performance of Treemap was desirable for tree overview and comparison tasks. Space Tree was most preferred over Treemap and Icicle, in terms of searching, counting and interactions, while hyperbolic tree was least preferred. Experiments also suggested the use of effective interaction mechanism with the touch device e.g. touch-events, to avoid user’s frustration.

7.2 Future Work

Various improvement and enhancement can be incorporated to make the WebVis tool more efficient and useful for educational purposes. Few of the initial requirements are listed below:

- Provision for providing a better user friendly interface for querying stored back-end data.
- A well structured relational data with inclusion of all data integrity constraints should be developed to be accessed from the tool.
- Query mechanism can be further improved that should include other sort of users’ queries e.g. Number of records to be fetched, fetching of specific records.
- Layouts can be modified to provide data editing and deleting and same should be reflected at the back-end.
- Various new data interaction techniques can be included in the layouts e.g. mapping of multiple parents to a single node, drag and drop sub trees.
• Various browser based application can be built on top of these visualisation layout e.g. A hierarchical and more interactive view of Clicker quiz.

• Incorporating a user access and security mechanism to let only the authorised user to access and manipulate data.

• Improving database view on the last slider, to be more understandable and logical for the viewer.

• Modifying file upload mechanism by providing support to various other file formats for data uploading.
Bibliography


