Hierarchical-Layout Treemap for Context-Based Visualization

Jingtao Wang[1], Yinwen Shen[2], Dingke Kong[2], and Jing Hua[2(✉)]

[1] School of Computer Science, Harbin Institute of Technique, Harbin, China
[2] School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
xwang@zjgsu.edu.cn

Abstract. In order to represent hierarchical information more efficiently and aesthetically on mobile devices, many researchers have adopted the space-filling method—Treemap. In this article, a novel dynamic hierarchical-layout algorithm is proposed to visualize hierarchical information on mobile devices. This layout algorithm tries to represent the hierarchical data set with more balanced aspect ratio for the rectangles which is closely related to the aesthetics of treemap. Furthermore, a focus+context fisheye distortion algorithm is introduced to help users understand details by increasing the size of the interested items while decreasing the others. Experimental results show that the proposed method can provide a suitable resolution for visualizing hierarchical information with high efficiency and give focus+context views on mobile devices in real time.

Keywords: Treemap visualization · Hierarchy navigation
Dynamic context-based layout · Focus+context distortion

1 Introduction

As is well known, most information in the world is hierarchical information and the most common form of hierarchical representation is the so-called treemap, which is capable of representing hierarchical data [1]. It works by dividing the display area into rectangles whose areas correspond to an attribute such as a value or proportion of the hierarchical data set. Spatial positions between rectangles indicate the relationship of nodes. Treemap can make efficient display of space, thereby representing more hierarchical data sets. Now treemap has been widely applied in this world due to its advantages [2].

Previous methods on treemap can be categorized into two general categories: layout algorithms and interaction algorithms. The first category consists of the layout algorithms for browsing hierarchy structures through space-filling visualizations [1]. The second relates to the techniques that have been specially developed for browsing the details and the context in treemap.

1.1 Layout Algorithms

The original treemap layout algorithm uses parallel lines to divide a rectangle into smaller ones to represent the node and its children [1]. This method creates the layout

Z. Pan et al. (Eds.): Transactions on Edutainment XIV, LNCS 10790, pp. 27–39, 2018.
https://doi.org/10.1007/978-3-662-56689-3_3

that contains rectangles with high aspect ratio which leads to the loss of aesthetics and becomes hard to detect the details.

Shneiderman and Wattenberg proposed a novel layout algorithm, namely ordered treemap layout [4]. This method keeps the original order of the data set and tries to address the instability occurred in the dynamically data updating.

But most work is not suitable for displaying quantum-sized objects, such as images. Benjamin B. Bederson, Martin Wattenberg and Dow Jones introduced the quantum treemap to accommodate the items such as photos [6].

Moreover existing treemap layout algorithms suffer from poor mapping between data order and visual order to some extents. Jo Wood and Jason Dykes generalized the existing squarified algorithm so as to exploit nodes more effectively to display geographical statistics data [7].

However, quite a small proportion of the space results in awfully small size of the rectangles due to the lack of details, which may become difficult to be recognized [3]. Therefore, this motives the interaction for treemap.

1.2 Interaction for Treemap

The interaction for treemap explores the set of hierarchical data in-depth, including the basic operations and the distortion methods.

The basic interaction for treemap is drill-down and roll-up. Drill-down and roll-up navigate the further or former level of the node which gives the detail about its children or father [12].

Compared to the basic interaction, Liqun Jin and David C. Banks extended See-Through interface proposed by Bier and Stone's [7] and introduced an interactive system called TennisViewer. It is used for visualizing the dynamic and tree-structured tennis match data [8]. They adopted Magic Lenses to explore the information at lower layers which can be laid atop each other easily to produce deep zooming.

Keahey and Visual Insights, Inc. described how focus+context techniques can be integrated into other high-level visualization paradigms [9].

Furthermore, a distortion algorithm based on fisheye and continuous zooming techniques is introduced to browse the data in treemap representation by Shi, Iranni and Li [10]. Their distortion algorithm increases the size of the interested node while shrinking its neighbors.

Ying Tu and Han-Wei Shen presented a seamless multi-focus and context technique, called Balloon Focus. It allows users to smoothly enlarge multiple treemap items as the focus, while maintaining a stable treemap layout as the context [11].

In this article, a novel context-based hierarchical-layout algorithm is introduced to represent the hierarchical information on mobile devices more effectively. Furthermore, a focus+context distortion algorithm is proposed to help users understand the items.

2 Materials and Methods

In this section, a novel context-based hierarchical layout algorithm which extends the ordered layout algorithm [5] is presented in detail. The algorithm explores better aspect ratio of the rectangles by splitting the total value into one or two parts. Meanwhile, a

novel focus+context distortion algorithm imitating fisheye is proposed in the way of increasing the interest and shrinking the others.

2.1 Data Structures

Each item in treemap data sets represented hierarchically as father and children like tree. Each node must contain enough information such as its size and label in order to represent the details of the item.

A node is defined by the following information:

- **Label (name):** it represents the name of the node, such as the menswear of a dress.
- **Size (weight):** it represents the size or the weight of the node in data set. It may be the size of the file for a data file or the value of a certain product sale.
- **Parent:** it points to the parent node.
- **Children:** this field contains an array of pointers to each of its children.
- **Width, Height:** the width and height of the rectangle influence the aspect ratio as it is drawn on the display area. These values are computed according to its size and the width and height of the display area.

In addition, this article also introduces several important data items for the distortion algorithm. The constant MIN_SIZE represents the smallest size that the node should increase to maintain enough information. The variable—sign will denote the number (one or two) of rectangles to be drawn, and is determined by the layout algorithm and the node size. A variable flag is employed to represent the orientation of the rectangle.

2.2 Hierarchical-Layout Treemap for Context-Based

The following layout algorithm extends the work by Shneiderman and Wattenberg [4]. It adopted the constant DIFF to determine the number of the rectangles drawn at the same time. Through this method, the algorithm gives the advantageous block aspect ratio for representing hierarchical information on mobile devices. The procedure of layout algorithm (see Fig. 1) will be described below.

Step 1: Sort the nodes
Sort the nodes in decreasing order according to the size (value or proportion) of each item in data set at the same level.

Step 2: Divide the nodes into two arrays
The function getPerfectSplitNumber is defined to search the advantageous split number of the blocks drawn at the same time. Since the nodes have already been sorted in Step 1, the nodes are further divided into two partitions: ArrayA and ArrayB, which contain the biggest node and the remaining notes respectively.

Step 3: Get the advantageous split number
The ratio denotes the division of the value in ArrayA by the total value at the same level. Then HEIGHT and WIDTH are used to set to the width and height of the ArrayA according to the following criteria.

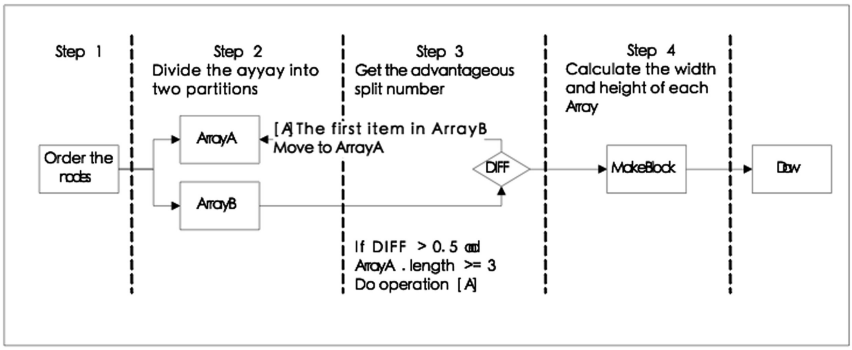


Fig. 1. The procedure of the proposed layout algorithm

$$\begin{cases} width = WIDTH, height = HEIGHT * ratio \\ when\ WIDTH < = HEIGHT \\ width = WIDTH * ratio, height = HEIGHT \\ when\ WIDTH > HEIGHT \end{cases}$$

The algorithm assists the division of width by height to the variable of ratioWH and the division of height by width. Then DIFF is calculated as follows.

$$\begin{cases} DIFF = 1 - ratioWH\ when\ width < height \\ DIFF = 1 - ratioHW\ when\ height > = width \end{cases}$$

If DIFF >0.5 and the length of the array ArrayA >= 3 then the advantageous split number is set to 2; otherwise 1. This means that the number of the rectangles which are drawn at the same time is 2 or 1.

Step 4: Set the height and the width for both arrays (rectangles)
If DIFF is more than 0.5, move the first node of ArrayB to ArrayA and calculate the width and height for ArrayA and ArrayB. Then Step 3 is performed recursively to calculate the advantageous split number.

Step 5: Draw rectangles
When the split number of the rectangles is 1, draw the rectangle with the width and height assigned by the method in Step 3 until all rectangles are finished.

2.3 Focus+Context Distortion

The distortion algorithm increases the size of the interest node and decreases the size of the others. To keep the original position of the node when distorted, the variable flag is introduced to represent the orientation of the rectangle and is combined with the advantageous split number. When flag equals 0 (or 1), it means the orientation of the rectangle is horizontal (or vertical). The constant MIN_SIZE donates the minimum size

of the interested node to be increased when the size is less than MIN_SIZE. Otherwise, the size will be increased to three times of the original size.

The distortion algorithm gives the focus+context view of the interested nodes. This interaction is activated when the finger moves on the focus and is deactivated when the finger moves away the touch screen. The algorithm will efficiently find the corresponding node as the finger moves on the region. The procedure is described as follows.

Step 1: achieve the orientation of the rectangles
The orientation of the rectangles is determined in the process of the width and the height setting. Setting the width of the rectangle to WIDTH and the height to HEIGHT*ratio when WIDTH is less than HEIGHT means the orientation of the rectangle is determined by the value of the flag. The value of flag is 1 represents the vertical orientation when WIDTH is bigger than HEIGHT.

Step 2: catch the split number of the nodes at the level
The split number and the orientation determine the layout of all nodes at the same level. They guarantee the order and layout of all nodes when distorted.

Step 3: calculate the size of the interest node and the others
The default size that the interested node will be increased is three times the original size. But the distortion size must be at least the MIN_SIZE.

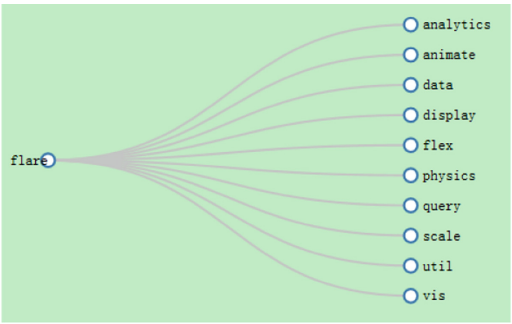
At last, the focus+context fisheye distortion works.

3 Experimental Results

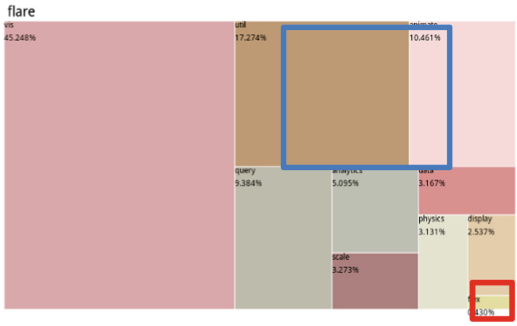
The layout algorithm and the focus+context fisheye distortion algorithm are applied on the Android platform. The data set is constructed according to the hierarchical data storing in JSON document which is known as a lightweight data interchange format based on JavaScript. Some experiments are implemented as follows.

Experiment 1 shown in Fig. 2 was designed to display the weight of each part which makes up the visualization such as util, query and animate. The JSON document constructs the data set as the hierarchical data with multi-level of father and children. The labels including name and weight of the rectangle are displayed to give the details about the nodes.

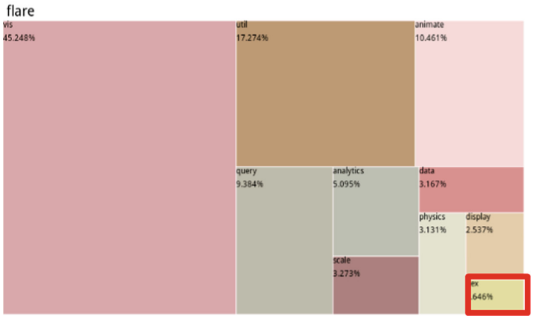
The data set about composition of visualization methods is represented by the familiar hierarchical visualization—Tree, which has the drawback of low space usage in Fig. 2(a). Compared to the Tree, treemap describes the nodes with the attributes of name and proportion in Fig. 2(b). It shows the full-usage of screen rather than Tree. In Fig. 2(b), some rectangles may be too small to understand the details for the reason of the smaller proportion especially in the bottom right corner surrounding the red box. When the focus moves to that rectangle, the focus+context fisheye distortion algorithm will work, as shown in Fig. 2(c). The size of the rectangle increases to three times the original size as presented by the red box in Fig. 2(c). So users can quickly catch the details of the interested node including the name and the weight.



(a) data set at the first level



(b) normal view of the treemap

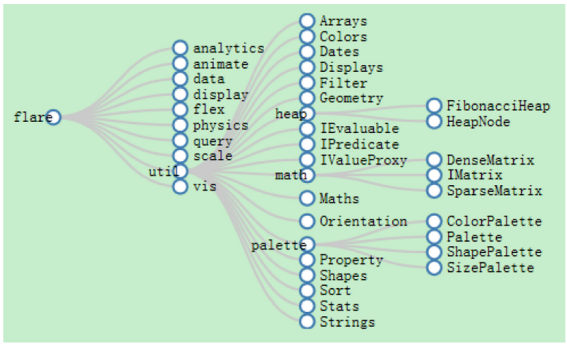


(c) distortion view of the treemap

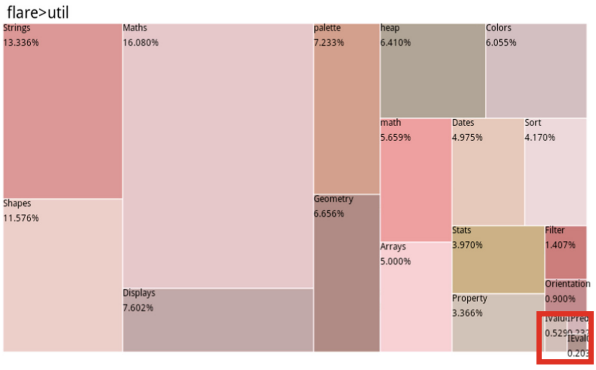
Fig. 2. Treemap layout and focus+context distortion (Color figure online)

Drilling-down by finger touching on the region of the rectangle is also implemented.

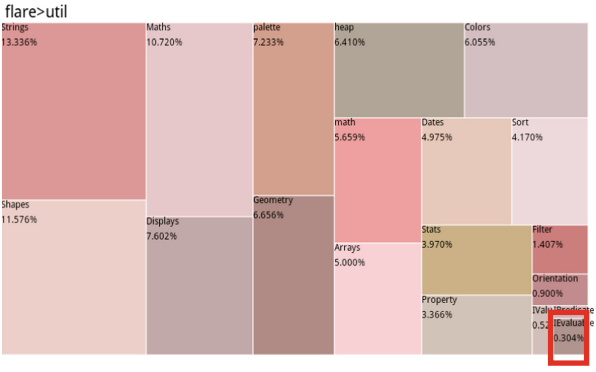
The second experiment is also aimed to represent the actual sales data from taobao.com by layout method and focus+context distortion algorithm on mobile devices. Part of the data set is described in Fig. 4(a) by Tree. Figure 4(b) and (d) demonstrate the data with red circle and blue circle respectively. These two charts have the same



(a) children of node UTIL

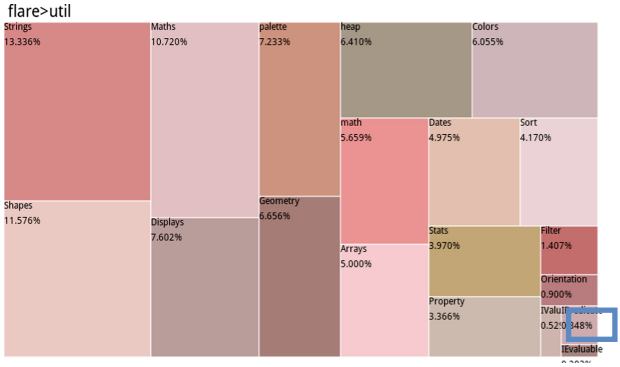


(b) the structure of UTIL



(c) normal view

Fig. 3. Visualization and distortion of node UTIL (Color figure online)



(d) distortion of node PREDICATE

Fig. 3. (*continued*)

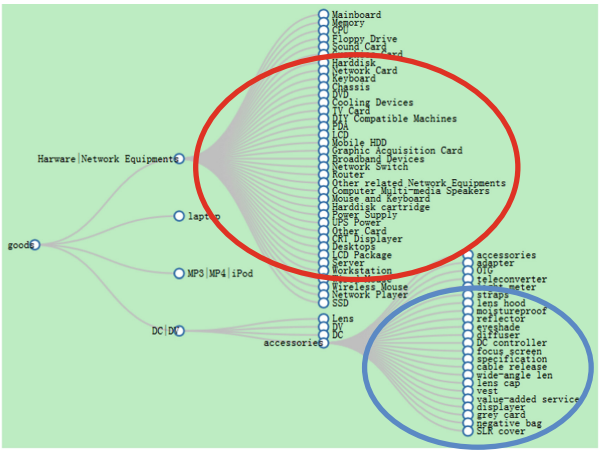
characteristic that the rectangles in the bottom right corner are hard to select, understand and divide. Figure 4(c) demonstrates the focus+context fisheye distortion of node photographic accessories with the advantages of easily selecting and understanding the details about the interest node in red box. Figure 4(e) does the same work as Fig. 4(c) in blue box.

The third experiment is still used to demonstrate the report of the infectious diseases in China, which is released by the State Statistics Bureau. The table of data set is divided into two parts according to the type of the infectious diseases that contain the Class (A and B) infectious diseases and infectious disease C. The construct of the data is represented by Tree in Fig. 5(a).

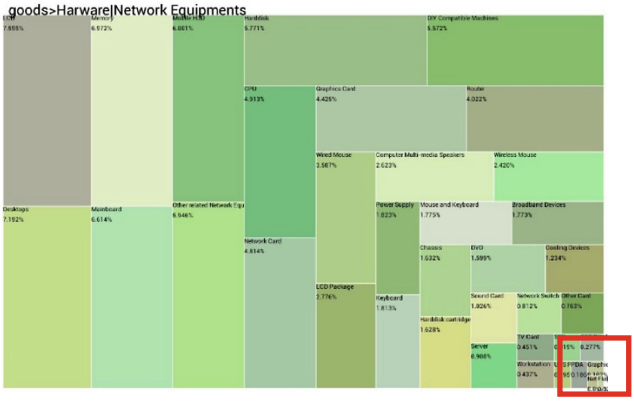
The second experiment is also aimed to represent the actual sales data from taobao.com by layout method and focus+context distortion algorithm on mobile devices. Part of the data set is described in Fig. 4(a) by Tree. Figure 4(b) and (d) demonstrate the data with red circle and blue circle respectively. These two charts have the same characteristic that the rectangles in the bottom right corner are hard to select, understand and divide. Figure 4(c) demonstrates the focus+context fisheye distortion of node photographic accessories with the advantages of easily selecting and understanding the details about the interest node in red box. Figure 4(e) does the same work as Fig. 4(c) in blue box.

The third experiment is still used to demonstrate the report of the infectious diseases in China, which is released by the State Statistics Bureau. The table of data set is divided into two parts according to the type of the infectious diseases that contain the Class (A and B) infectious diseases and infectious disease C. The construct of the data is represented by Tree in Fig. 5(a).

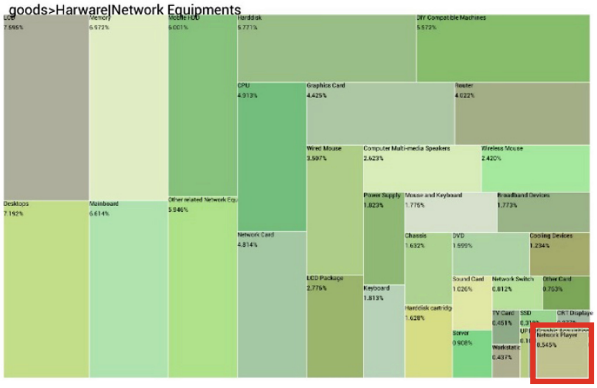
The experiment with hierarchical-layout algorithm is presented in Fig. 5(b) and (d). This layout sorts the number of the death and sets the advantageous aspect of the



(a) composition of Tree

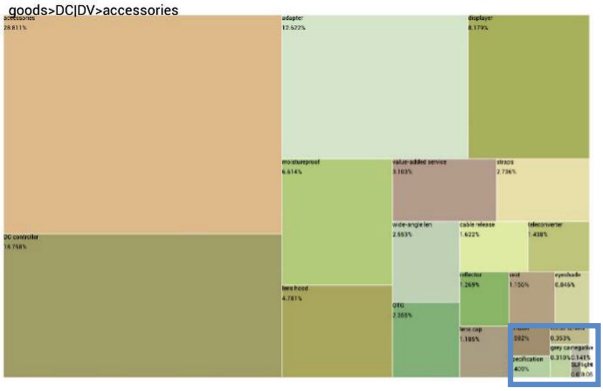


(b) normal layout view

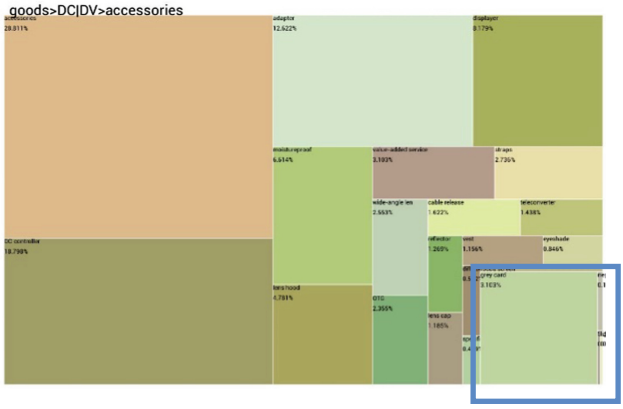


(c) distortion of PHOTOGRAPH

Fig. 4. Visualization and distortion presentation (Color figure online)



(d) normal view of LIGHT

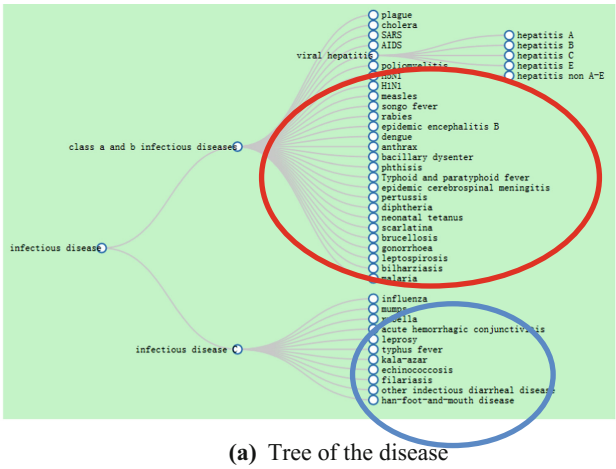


(e) distortion of LIGHT

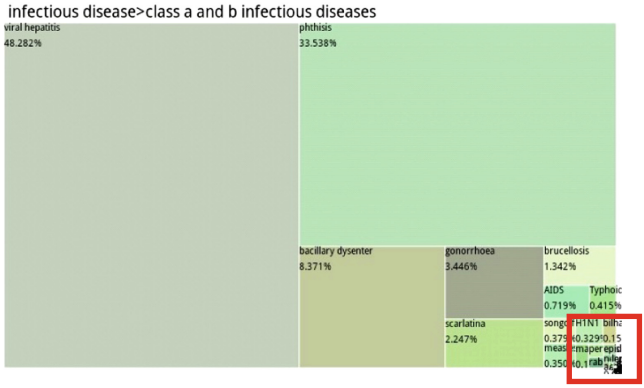
Fig. 4. (*continued*)

rectangles for the nodes. In Fig. 5(b), users can easily discover the drawback caused by the layout: the bottom right corner of the display area looks crowded and it is hard to select the red box. Figure 5(c) solves this problem by the focus+context fisheye distortion algorithm. The size of the interested node of epidemic encephalitis B is increased to three times the original size in Fig. 3(c) with red box. Figure 5(e) also does the same work as described in Fig. 5(c).

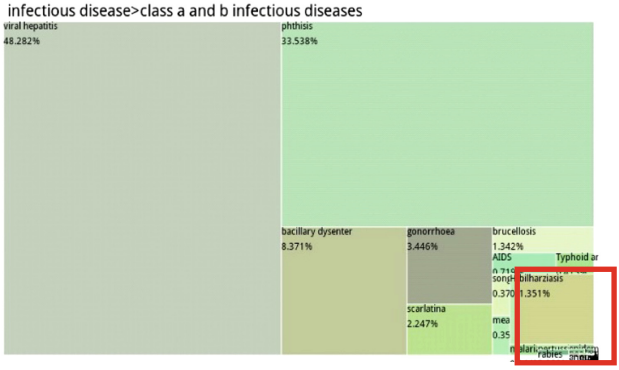
The three experiments on Android platform are designed to evaluate the effectiveness and aspect ratio of rectangles by the novel methods on mobile devices. The results suggest that the subjects are faster at browsing and locating objects of interests in the layout. The effect will not be influenced even when there are a large number of nodes at the same level.



(a) Tree of the disease

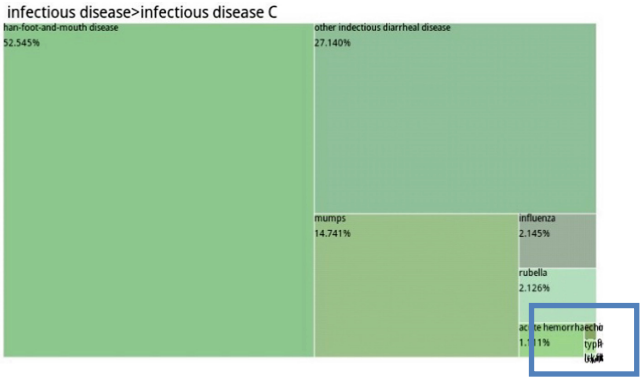


(b) unclear layout view

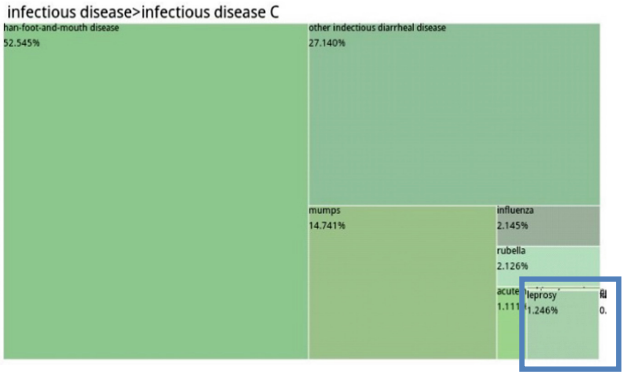


(c) distortion of BILHARZIASIS

Fig. 5. Comparison between tree and treemap and distortion of interest (Color figure online)



(d) distortion of RUBELLA



(e) distortion of LEPROSY

Fig. 5. (*continued*)

4 Conclusions and Future Work

The novel dynamic hierarchical-layout algorithm of treemap has been introduced to efficiently and aesthetically represent the hierarchical information on mobile devices as on PCs. This method has better layout performance with balanced aspect ratio which is closely related to the ascetics of treemap. Furthermore, the fisheye distortion algorithm helps users understand the details of the items deeply. This technique simulates the fisheye distortion in the way of increasing size of the interest and decreasing the other. This method gives a real-time focus+context view with the smooth transformation between the normal view and the distortion view.

Additional work under consideration is to improve the layout algorithm to better suit different data set which may cause the disadvantageous aspect ratio. The layout algorithm should enhance the adaptive ability to the distinct pieces of data.

In addition, another variation of the distortion is to consider more than one node distortion at the same time. The further work can be organized as the nodes in the

circular region with certain radius increase. In this paper, only the linear distortion is considered, the nonlinear distortion of the fisheye will be investigated.

Acknowledgement. This work was supported in part by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (No. 2014BAK14B01), National Natural Science Foundation of China (No. 61379075), Zhoushan Municipal Science and Technology Plan Project (No.142014C21035). The authors are grateful for the anonymous reviewers who made constructive comments.

References

1. Shneiderman, B.: Tree visualization with treemaps: a 2D space-filling approach. ACM Trans. Graph. **11**, 92–99 (1992)
2. Jungmeister, W.A., Turo, D.: Adapting treemaps to stock portfolio visualization. Technical Report CS-TR-2996. University of Maryland, College Park, USA (1996)
3. Bruls, M., Huizing, K., van Wijk, J.: Squarified treemaps. In: Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, 29–30 May 2000, Amsterdam, Netherlands, pp. 33–42 (2000)
4. Shneiderman, B., Wattenberg, M.: Ordered treemap layouts. In: Proceedings of the IEEE Symposium on Information Visualization, 22–23 October 2001, San Diego, CA, USA, pp. 73–78 (2001)
5. Wood, J., Dykes, J.: Spatially ordered treemaps. IEEE Trans. Visual. Comput. Graphics **14**, 1348–1355 (2008)
6. Bederson, B.B., Shneiderman, B., Wattenberg, M.: Ordered and quantum treemaps: making effective use of 2D space to display hierarchies. ACM Trans. Graphics **21**, 833–854 (2002)
7. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: The see-through interface. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, 2–6 August 1993, Anaheim, CA, USA, pp. 73–80 (1993)
8. Jin, L., Banks, D.C.: TennisViewer: a browser for competition trees. IEEE Comput. Graphics Appl. **7**, 63–65 (1997)
9. Keahey, T.A.: Getting along: composition of visualization paradigms. In: Proceedings of the IEEE Symposium on Information Visualization, 22–23 October 2001, San Diego, CA, USA, pp. 37–40 (2001)
10. Shi, K., Irani, P., Li, B.: An evaluation of content browsing techniques for hierarchical space-filling visualizations. In: Proceedings of the IEEE Symposium on Information Visualization, 23–25 October 2005, Minneapolis, MN, USA, pp. 81–88 (2005)
11. Tu, Y., Shen, H.W.: Balloon focus: A seamless multi-focus+context method for treemaps. IEEE Trans. Vis. Comput. Graphics **14**, 1157–1164 (2008)
12. Turo, D., Johnson, B.: Improving the visualization of hierarchies with treemaps: design issues and experimentation. In: Proceedings of the IEEE 3rd Annual Conference on Visualization, October 19–23, 1992, Boston, MA, USA, pp. 124–131 (1992)