A Magic Treemap Cube for Visualizing Olympic Games Data

Xiaowu Chen, Haolin Yang, Yongtao Ma and Bin Zhou

State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract—**Treemap, a common information visualization approach for dividing a display area into nested rectangles, has a good performance on visualizing hierarchical data. And it enables user to compare nodes in the same level and to learn certain relationship between adjacent levels. However, treemap layout is limited to visualize multi-dimensional hierarchical data in which the hierarchy is unfixed and complex, especially when the hierarchy is an arbitrary graph. Considering this problem, we present a novel extension called magic treemap cube, which improves the layout and has a good performance on visualizing unfixed (cross-level) and complex data, inverting hierarchy without reorganizing the structure, and enabling users to easily compare among treemaps. The magic treemap cube is a three-dimensional regular hexahedron whose six faces are covered by six different treemaps. Each dimension of the magic treemap cube is a hierarchy structure specified by users according to a hierarchical division of visualization information. With the treemap algorithm, tagcloud and mapping rules, the treemaps are automatically controlled by permutations of three dimensions of the cube. Users could observe the six correlative treemaps respectively by rotating the magic treemap cube in three dimensional Cartesian coordinate. And this magic treemap cube can be unfolded into a tiled treemap in 2D Cartesian coordinate for users-friendly comparison of each treemap. Meanwhile, a merging operation to the tiled treemap is proposed to not only compare two treemaps through the adjacency relations among faces but also provide more information. For demonstrating the above characteristics, an application (called OlympicVis) has been implemented to visualize the Beijing 2008 Olympic Games information, which involves multi-dimensional hierarchy and interconnected data. The OlympicVis can display Cube View, Schedule View, Medal View, Date Filter and Time Line.**

Index Terms—**treemap, hierarchy visualization, information visualization, treemap cube.**

I. INTRODUCTION

The Olympic Games information, such as game schedules and medal winners, involves multi-dimensional hierarchical data in which the hierarchy is unfixed and complex. The information is usually described in text or tables on the websites. Although the hierarchy visualization, such as treemap, has a good performance on visualizing hierarchical data, it is mostly limited to visualize multi-dimensional hierarchy and interconnection of the Olympic Games information, especially

E-Mail: chen@buaa.edu.cn

when the hierarchy is an arbitrary graph. Moreover, in order to find the favorite Olympic Games information, people have to spend much time and energy to look for, read and understand the text step by step.

Information visualization is the interdisciplinary study of the visual representation of abstract data sets, such as unstructured texts or points in multi-dimensional space [1]. It utilizes graphical techniques to help people understand and analyze data, and provides deep insight into the structure of data [2]. One of the most common relations among abstract information is hierarchical relation, which is almost ubiquitous and appeared in many different applications. Hierarchical data is of a data model, where data is organized into a tree-like structure using parent/child relationships [3]. Usually each parent can have many children but each child has only one parent in this structure.

Treemap [6] is a common information visualization approach for displaying hierarchical (tree-structured) data, dividing a display area into nested rectangles. Each branch of the tree structure is assigned with a rectangle, which is then divided into smaller rectangles representing sub-branches. Each leaf has an area size proportional to the dimension of the hierarchical data. Treemap enables user to compare nodes in the same level and to learn certain relationship between adjacent levels. And since the area size is correlated with the tree structure, one can easily see the patterns that would be difficult to spot in other ways.

Although treemap has good performance on visualizing hierarchical data, the visualization results occupy a large amount of space. It is difficult for user to observe one certain node or the entire structure, and it is not easy to change the hierarchy (hierarchical relation) without reorganizing the structure. Moreover, the Olympic Games information typically involves multi-dimensional hierarchical data which is unfixed and complex, especially an arbitrary graph. However, the treemap layout is limited to visualize this information.

Thus, we present a novel extension called magic treemap cube. The magic treemap cube is a three-dimensional regular hexahedron whose six faces are covered by six different treemaps. Each dimension of the magic treemap cube is a hierarchy structure specified by users according to a hierarchical division of visualization information. With the treemap algorithm, tagcloud and mapping rules, the treemaps are automatically controlled by permutations of three dimensions of the cube. Users could observe the six correlative treemaps respectively by rotating the magic treemap cube in three dimensional Cartesian coordinate. And this magic

Fig. 1. A demonstration of our magic treemap cube for visualizing Olympic Games data. The left shows the cube controlled by three dimensions and covered by six different treemaps. Each dimension is a hierarchy specified by users according to a hierarchical division of visualization information. The treemaps are displayed in the form of permutations of cube's three dimensions with treemap algorithm, tagcloud and mapping rules. The middle shows the treemap covering one face of the cube. Users observe the six correlative treemaps respectively by rotating the cube in three dimensional Cartesian coordinate. The right shows the tiled treemap. The magic treemap cube can be expanded in 2D Cartesian coordinate.

treemap cube can be unfolded into a tiled treemap in 2D Cartesian coordinate for users-friendly comparison of each treemap. Meanwhile, a merging operation to the tiled treemap is proposed to not only compare two treemaps through the adjacency relations among faces but also provide more information. There are two contributions for the magic treemap cube: 1) an improved layout of treemap to visualize unfixed (cross-level) and complex data; 2) a good performance, such as inverting hierarchy without reorganizing the structure, enabling users to easily compare among treemaps.

The remainder of this paper is organized as following: Section 2 introduces the related work. Section 3 presents the novel extension in the form of magic treemap cube for visualization. In Section 4, we discuss an application (called OlympicVis), which visualizes the Beijing 2008 Olympic Games information using the magic treemap cube. Finally, conclusion is given in Section 5.

II. RELATED WORK

In this section we briefly present some of the previous work related to visualization of hierarchical information and presentation/visualization of Olympic Games data.

2.1 Hierarchical Information Visualization

Hierarchical structure is ubiquitous in information organization. Visualization approaches for hierarchical structure include treemaps, cone trees, hyperbolic trees, beamtrees and radial visualizations [4]. The treemap visualization technique [6], developed by Ben Shneiderman, maps hierarchical information to a rectangular 2D display in a space-filling manner so that 100% of the display area can be utilized.

In treemap method, the layout algorithm divides the display area into smaller rectangles corresponding to the child nodes, and the area of each rectangle is determined by the weights of the nodes. When the treemap was first introduced, the slice-and-dice algorithm [2] was the only layout algorithm, dividing rectangles by parallel lines. Examples of treemap using slice-and-dice layout algorithm include [5][6], which

visualize the contents of a hard disk with leaf nodes having areas corresponding to file sizes.

Treemap has since been widely adopted and applied to a various problems. For example, treemaps have been applied to visualize program execution data in Wattenberg's visualization of stock market data[7], to present groups of related pictures in photo-browsing applications[8], to visualize Usenet activity [9][10], and to visualize currently popular items on the social bookmarking site del.icio.us[1].

As shown in Fig. 2, in the original treemap approach which visualizes an organizational chart, it is difficult to compare the size of different nodes within the results [11]. Thus Van Wijk and Van de Wetering propose the cushion treemap by shading a treemap [11]. Cushion treemaps can use the display area effectively without any additional pixels, but they have been found difficult to reliably interpret [4][12].

Take the work by Demian and Fruchter on Corporate

Fig. 2. Van Wijk and Van de Wetering's example of treemaps degenerating to a uniform grid for a large balanced tree [11].

Memory [13] as an example to demonstrate the structure of a nested treemap. The hierarchy is presented using a combination of varying line thickness, padding and labels between nodes. Although labels at different levels in the hierarchy are given

---

[1] del.icio.us Most Popular Treemap. http://codecubed.com/map.html.

different colors and placements, it is still difficult to attach labels on one level of the hierarchy [4].

2.2 Olympic Games Data Presentation or Visualization

The Olympic Games is one of world-famous grand meetings held every four years. There are more and more people interested in this game. The Games of the XXIX Olympiad took place August 8-24, 2008, in Beijing, China. Approximately 10,000 athletes from 204 countries and regions participated in this game with around 20,000 accredited media bringing the Games to the world, and it is concerned about by an audience of 4.5 billion around the world. In the Beijing 2008 Olympic there are 32 sports and every sport contains several events, which sum to 302 events in total.

Fig. 3. Olympics Medals by Event in Many Eyes using treemap, providing a series of optional views by adjusting the order of four options: event, county, medal, and discipline (sport) [14].

Many Eyes Olympic Medals Visualization. Many Eyes, launched by IBM Research, is a public web site where users may upload data, create interactive visualizations, and carry on discussions. The goal of Many Eyes is to support collaboration around visualizations at a large scale by fostering a social style of data analysis in which visualizations not only serve as a discovery tool for individuals but also as a medium to spur discussion among users. To support this goal, the site includes novel mechanisms for end-user creation of visualizations and asynchronous collaboration around that visualization [14].

As Fig. 3 shows, one researcher uploads some Olympic Games data, and creates a visualization application for Olympics Medals using Many Eyes. There are a series of views in this visualization application, but the four views display the same information actually. And the information only mentions medals, so it's hierarchy too single to make Users find the relation between sports and events, and even understand the comparison among the information. There are also no time lines in Many Eyes, there for users have no idea about the time information in Olympic Games.

Nytimes Olympic presentation. Nytimes [2] website is a portal of news site in America. During the Olympic Games, It provides an event and schedule presentation through using calendar.

Nytimes Olympic presentation is very practical. It could display simple interconnected relation, such as, between event and schedule. However, since it uses the method of calendar,

---

the presentation is lack of the relation between nations, sports and medals. It presents the static sport schedule and dynamic news of Olympic Games only.

III. MAGIC TREEMAP CUBE FOR VISUALIZATION

3.1 Visualization Pipeline and Dataset

The magic treemap cube based visualization application includes three main parts: data collector, data formatter and visualization tool, as Fig. 4 shows.
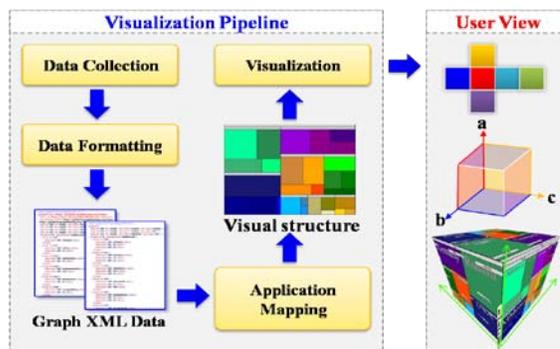
Fig. 4. The visualization pipeline and user view of OlympicVis application.

While taking the Olympic visualization as an example, the Olympic dataset is one of the main parts of OlympicVis, and it is the input source of visualization tool. About Beijing 2008 Olympic Games, there are several types of information: schedules, sports, events, nations, and medals. Schedules are relative to sports and events, presenting the information about time and location of sports. Medal information is concerned with athletes, nations and events. On nation information, we focus on what type of medal (Gold, Silver and Bronze) the country gets in which event of sport. The above information involves multi-dimensional hierarchy and interconnected data as Fig. 5 shows.
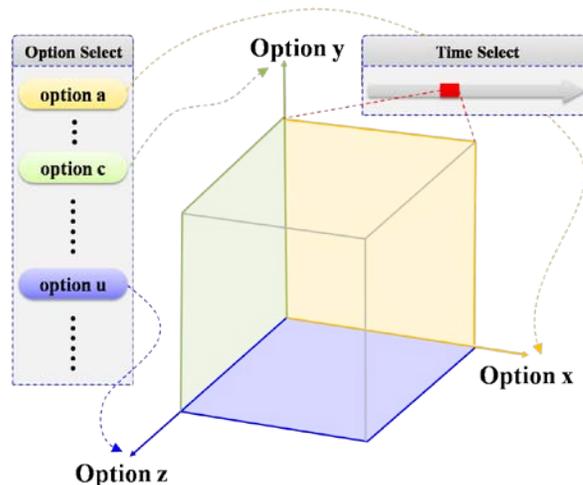
Fig. 5. Interconnected data or information (indicated the relations) in applications.

In order to generate multi-dimensional hierarchical data, data collecting and data formatting are the two important steps. Data collector that we use for the first step is a web crawler to obtain the relevant text information in html pages. Then we need to format this information by Data formatter which is a XML analyzer and builder actually. When these two steps are completed, we can put the data into visualization tool to display.

3.2 Magic Treemap Cube

Treemap has a good performance on visualizing hierarchical and interconnected data, which enable users to compare nodes in the same level and learn certain relationship between adjacent levels. However, the layout still has several limitations. At First, treemap layout is incapable of visualizing data in non-adjacent level. For example, a treemap can not present relationship between nodes in level n with nodes in level n-1 and level n-2 at the same time. Secondly, given hierarchical data, the hierarchy cannot be inverted unless users recognize the data manually. Besides, previous works have to ignore the importance of comparison among two or more treemaps which may provide additional information than only one treemap. These limit the inflexibility of treemap when it visualizes hierarchical data. We propose a novel method to allow traditional treemap to visualize cross-level data, invert hierarchy without reorganization of the data, and enable users to easily compare information among multi treemaps.

We utilize cube to improve current treemap layout to achieve this goal. This method has considerable merits, lies in three respects.
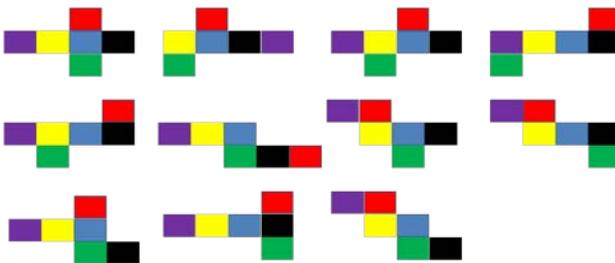
Fig. 6. 11 distinct nets for the cube.

First, notice that each face of a cube is decided by its relevant two axes, which indicates that users can custom the hierarchy if they associate an axis with data in certain level. For example, if x-axis is associated with nodes in level n, and y-axis is associated with nodes in level n+i, then the treemap in x-y face will reflect the relationship between nodes in level n and level n+i. In this way visualization of cross-level data can be achieved, which is especially applicable to the circumstance that each two levels are interconnected and the hierarchy can be generated automatically by application. Moreover, we should also notice that we can define 6 hierarchies by setting three axes, which will significantly simplify user interaction.

Second, the treemap cube can automatically reflect inverted hierarchy. The cube has a prismatic symmetry, which can be represented by vertex-transitive coloring the faces. Cube with sides alternating colors, both the opposite sides has the same

color. It indicates that the treemap cube can visualize a hierarchy in one face and the inverted hierarchy in the opposite side.

Finally, the treemap cube performances well when it is applied to compare among multi treemaps. Notice that adjacent sides of a cube share a same axis, which indicates treemaps on adjacent sides share the same level of data and comparison of those adjacent treemap may further provide potential information. To better support the comparison of treemap, the treemap cube can be unfolded. It has been proved that there are in total 11 distinct nets for the cube, as shown in Fig. 6. Each way of unfolding the treemap cube provides a distinct view to compare adjacent treemaps.

As Fig. 7 shows, the cube is covered by six faces, each of which can be considered as a treemap. You can rotate this cube to see the content of any face. If you want to see one face, you could expand this face to the bigger treemap visualization in order to see more clearly. There are three axes which control the display content of six faces. The arrangement of each axis has several alterations. For example, if users click one option in x-axis, the front, back, top, bottom faces should be updated according to the background data of the option concerned. After users click y-axis, the coordinate treemap could expand the 2-level information based on the x-axis treemap.
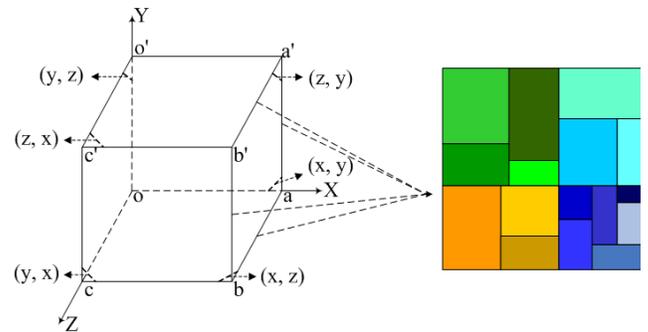

Fig. 7. Display the corresponding relationship between cube and treemap.

When it isn't convenient to observe the number of faces synchronous or when users want to compare several faces, we can unfold the cube as the Fig. 8 top center shows. Through the smooth animation, the cube can expand from 3D to 2D graphic display. The above process is also reversible. As shown in Fig. 8 top center, the face 1 is located in the cross center, which means the user wish to compare the other faces to face 1. If user designate another face as the cross center, the cross will change through defined animation until the designated face move to the center. Furthermore, beside the designated center face, the location relationship between other faces should be consistent with the original cube, as Fig. 8 top right shows.

In the cube-based structure, each face can only display two axes' content, which means that the treemap face can only display two-level nodes. Considering that users may need more information from the treemap since they want to know deeper level of information, the treemap mergence should be handy. According to relationship between faces and axes, we suppose that each face can be expressed in <x,y,z> partial order. Such as, if face2 is in the x-y Cartesian coordinate plane, its partial order is x-y. Suppose that face4's partial order is y-z, its first element
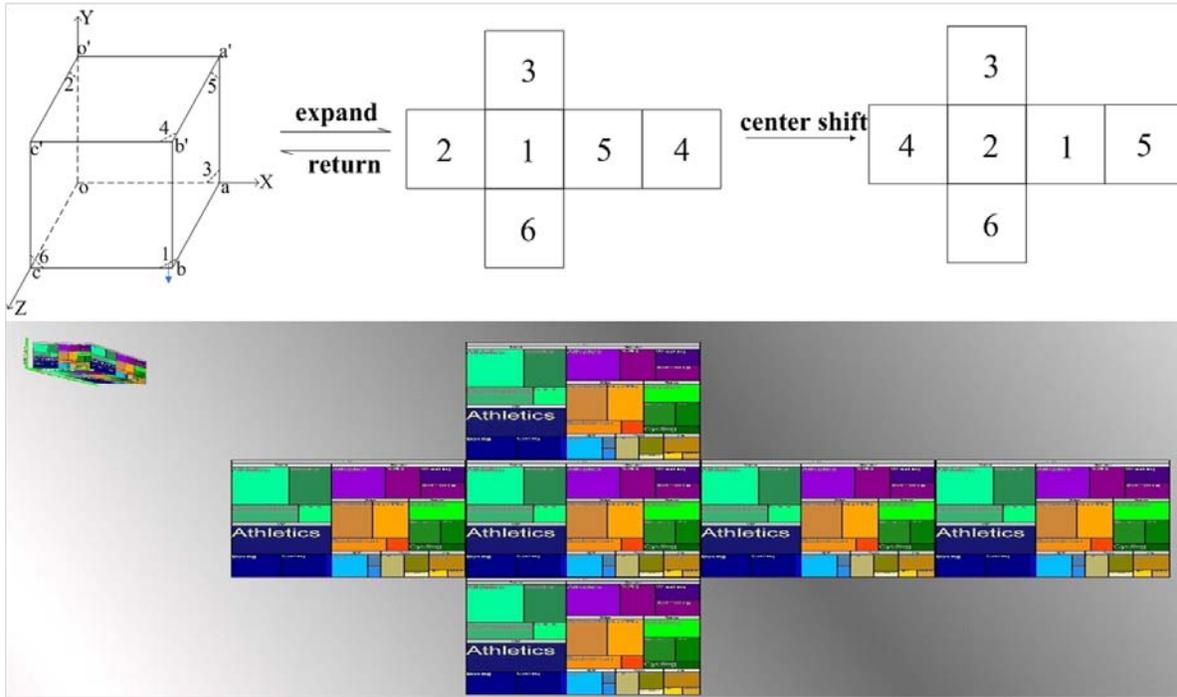
Fig. 8. The result of unfolding of treemap cube which enable user compare among it on adjacent sides

is the same with face2's second element, so these two treemap can be merged into a 3-level treemap, whose hierarchy is arranged in the following way: x is the first layer, y is the second layer and z is the bottom one. Thus we can conclude that

The process is similar to the treemap merging process discussed above, when users want to see relevant information in different time and compare through time. For example, if users want to see the information generated for the same or different objects in two days, we shall merge two days treemaps into one so the users can clearly understand the information. In order to make it easier for users to get information, the top node of the treemap could be formed by the same color scheme. The colors which display more brightness mean that these planes have greater weight.
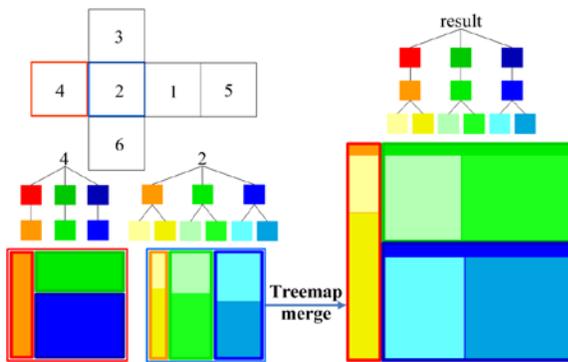
Fig. 9. Treemap merge schematic diagram.(Color Plate 1)

3.3 Visualization Algorithm

Magic Treemap Cube visualization algorithm is shown in the

there are six situations for these since our cube has six faces. As Fig. 9 shows, if the user wants to see the merged content of face 2 and face 4, they would merge these two 2-level treemaps into a 3-level.

above table. At first, with the XML which was collected from several website dataset, we can generate corresponding treemap. Secondly, when we know the position and size of the treemap's sub-elements --- face planes, we can splice these planes into the 3D treemap cube using flex pv3D technology. The detailed operations of the cube will be introduced in the section IV. Suppose users want to see a single treemap, they can interact with the cube to return to the single treemap observation.

Step 1: Generating related xml data from major portal websites;
Step 2: Using the dataset to generate the corresponding treemap;
Step 3: Extract treemap sub-element --- planes;
Step 4: Splice into treemap cube by planes using flex pv3D technology;
Through interaction with the cube, users can also back to Step 2 to observe the original treemap.

IV. VISUALIZING BEIJING 2008 OLYMPIC WITH MAGIC TREEMAP CUBE

4.1 The Cube Visual Structure and Views

The magic treemap cube proposed in this paper has a considerable merit when it is applied to visualized Olympic data, lies in two respects.

At first, the magic treemap cube significantly simplifies interaction. As discussed in section 3, items in Olympic dataset, such as nations, medals, schedules and sports are interconnected among every two or three items. The permutation of every two items is 12, which means if one user wants to observe all treemaps corresponding to every two items, he has to reset the treemap 12 times. Moreover, if additional

operations such as filtering are taken into account, the number of interaction may be even huger. To consider that, our cube cube can present 6 treemaps at the same time only requiring users set items of x, y and z axis, users only have to reset the treemap two times to view all the 12 treemaps. Besides, the filtering operation is also applied to all the 6 treemaps, which means such additional interaction can be further simplified.
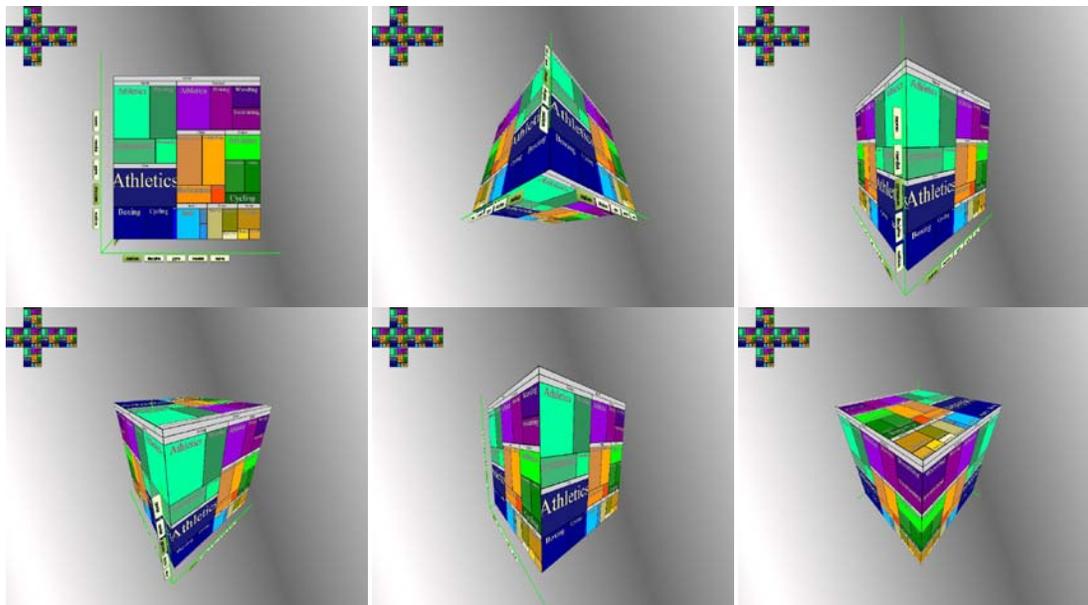
Fig. 10. To observe the treemap cube from six camera angles.

For the second, the magic treemap cube allows users to compare treemaps covering in all 6 faces. The comparisons among different treemaps can provide more information than by observing every treemap alone. For example, if the square size is to be decided by number of medals, from the nation-sport treemap in which the nation nodes are parents of sport nodes, we can infer that which nation in specified sport is more powerful; while from sport-nation treemap we may infer which nation dominate a certain sport. Unless these two treemaps are compared, it is doubt that whether we can draw any reliable and comprehensive conclusion of the competitiveness of a country in sports. As Fig. 10 shows, the users could observe the magic treemap cube in different angles to get the appropriate information.

4.2 The Treemap Visual Structure and Views

When unfolding the cube to treemap which we proposed in the above section, our methods generally provide three views to observe the treemap, one of which is the schedule view allows people to get information of the schedule through operating the cube. The second one is the medal count view from sport perspective which allows people to learn information of medal table through a combination of treemap and tagcloud. The last one is the medal count view from nation perspective, which is similar to the second one. The former two views will be discussed in detail as below.

1) The schedule view

The schedule view acts as a guide for the Olympic Games. Unlike the traditional schedule forms which are tables whose

rows represent items and columns represent dates, we displayed the schedule in a treemap. With the advancements of treemap, more information, for example, how many medals will be generated in a sport, can be deduced besides the schedule if we sizing the treemap with the number of medals of a sport. In Fig. 11 we can easily learn that swimming and athletics will generate the most medals. Moreover, treemaps concerned with sport, sized according to the number of medals and colored according to the number of related news in internet, can clearly demonstrate the scale and popularity of this kind sport.
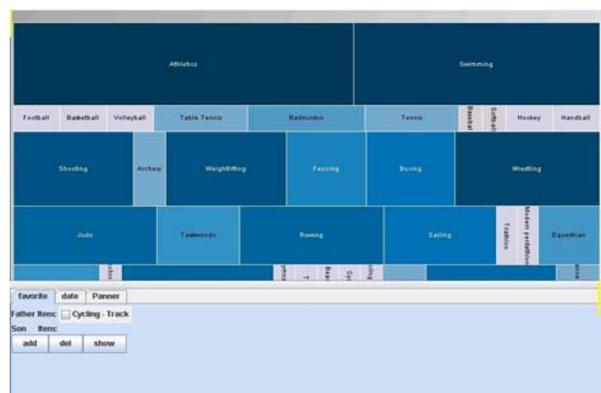

Fig. 11. Schedule view has been showing sports. In this picture, all the sports have been on the top level of hierarchical data.

There are almost 400 sports items (also called events). Displaying all these events in a limited area may make people ignore the information contained in the rectangles. For example,

swimming can be divided into more than 20 events so that it is impossible to show all the events in display area synchronous. Since the data is organized in hierarchical way, only data on the top level, such as swimming or archery, is displayed in the treemap initially. If one wants to know more, he could double-click on it to enable the applet to connect to the server and get detailed information, which will be displayed with the treemap afterwards. As Fig. 12 shows, the detailed information of the selected sport (such as powerlifting) is displayed, and we can learn about which event on the program of Powerlifting is
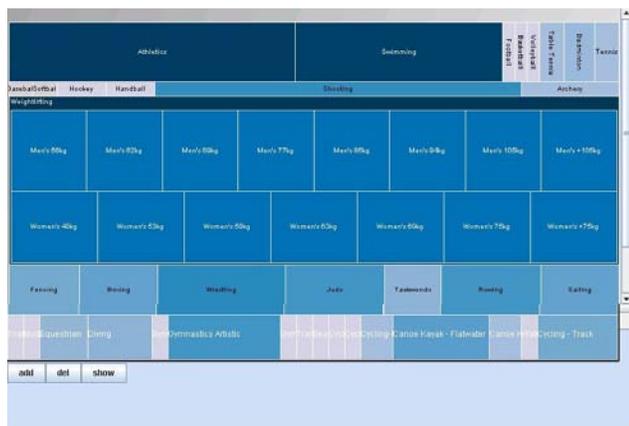
Fig. 12. Schedule view has been showing sports with detailed information. From this figure, we can learn that Men's-48kg, women's-44kg and so on are on the event of powerlifting.

Men's -48kg, Women's -40kg and so on.

Since treemap algorithm is not stable, every time when detailed information is loaded and displayed, the treemap may change a lot, and it provides an unfriendly change animation. Among the known treemap algorithms such as strip ,slice-and-dice and squarified, strip is finally choose to implement the Olympic treemap for the reason that it is relatively more stable than squarified algorithm and more square-like than slice-and-dice. However, since strip algorithm is not stable enough, and the squares may sometimes jump to another places abruptly, we design a method to smooth the unstable change, in which rectangles is moved from their start location to final gracefully by drawing animation. First of all, the distance between originate position and the end position is calculated. Then a moved path was found according to the two positions. Finally, we move the rectangle step by step. Through this way, the movement seems smoothly, and the instability of the treemap was decreased from the visualization.

2) The medal view

In previous method, medal may be concerned by tabular form most. Our work also uses treemap to represent medal and allows users to view medal count from two ways. One way is the common view, such as, nation-medal view. In this view, each square represent a nation and is sized according to the number of medals that the nation has gained. Furthermore, by double-clicking on a nation, the users may get information of in which sports that the nation gain medals, which is also displayed in treemap and sized according to the number of medals the nation gains. Through the nation-medal view we can easily learn the medal distribution among nations and which sport the nation is relatively good at.
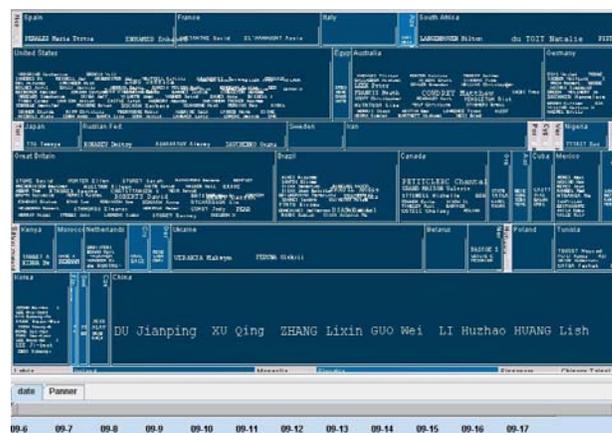

Fig. 13. The medal view, a combination of treemap and tagcloud. Information of Both medal table and medalists can be directly learned.

Moreover, users may not only be concerned about the medal table but also about medalists. To this end, we combine the treemap with text visualization. Tagcloud which is an appropriate method to visualize medalists that the more medals an athlete gains the bigger his tag will be. For example, Phelps won 8 gold medals in swimming and his name may be the easiest to catch at the first glance. Fig. 13 shows the medal view with tagclouds. Each tagcloud is limited in the related squares that represent the nation or sport it belongs to. However, the area of squares are unpredictable, thus our tagcloud should adapt to the area automatically. When the square is too small to draw all the tags in, we decrease the average font size of tags and erase athlete who win leaser medals. Otherwise, if the area is big enough, we enlarge the average font size of each tag on the premise that all the tag can be displayed. User can also scroll the mouse wheel to enlarge the display squares to see more and bigger tags.

4.3 Time Dimension

1) Date filter

Although we enable users to view what they are interested in to filter too many squares displayed in a limit area simultaneously, it may still happen if user selects too many sports. To avoid the situation, a date filter is designed and implemented; through a range slider that both left and right side is enable to slide to choose a date interval, to ensure sports that out of the date interval be erased from the treemap. Fig. 14 shows the result after being filtrated by date interval. Moreover, too many squares being displayed in a limited area may make the rectangles too small to display any information. Thus, the mouse wheel also allowed for user to enlarge the squares by rolling, and the unstable change triggered by doing so is also smoothed by defined animation.

2) Time line

The Beijing 2008 Olympic Games lasted for 17 days. In these days, there were thousands of matches for spectators to view. Then the Timeline view also acts as a guide for the Olympic Games. As a tool for users to conveniently inquire schedule information, unlike the traditional schedule displayed as a table, the Timeline's end-result is to enable the users who are not experience before, through some easy explanation, to

operate clearly as soon as possible.
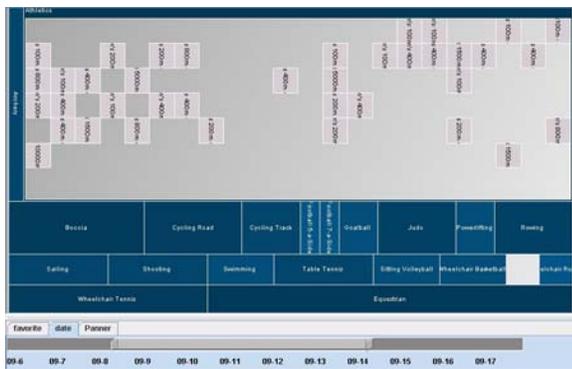
(a)
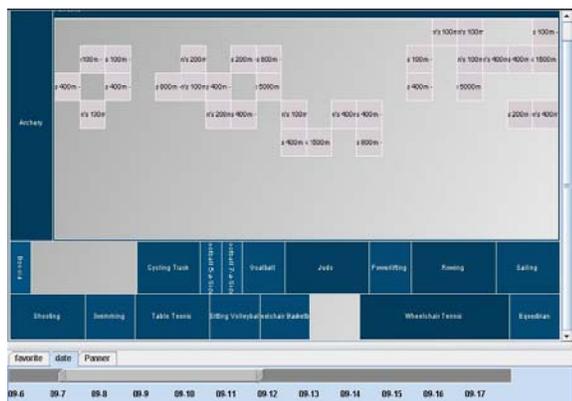
(b)

Fig. 14. Treemap with date filter. Graph (a) and (b) show results after different date interval are selected. The treemap with a date filter helps users to focus on what they are interested in.
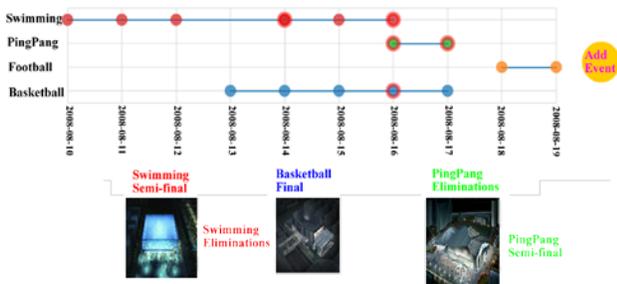

Fig. 15. User operate: users click the colorful nodes, then corresponding picture could display below the Timeline and the schedule information displayed in the form of text. Different information uses different color.

When users click event name in magic treemap cube, the Timeline scene will be popping. Fig. 15 top show a time table whose horizontal axis represents time nodes and vertical axis represents event name. The reason of designing in this mode is that through comparing the axes, the event and day information can be displayed clearly, then users can see clearly which node they want to choose. As the target for users to operate, every node in table represents selected event schedule information in specific day. If nodes belong to different event, they would be symbolized by different color to distinguish. In form of text, the schedule information whose color is similar with the

corresponding event is attached to the picture of gym. Then export the picture of venue into the scene, corresponding to the above-mention node, the "water cube" picture will be displayed. As for the animation for both picture display and text display is committed to allow users to experience more clearly.

Each node corresponds to its own venue, but the scene space is limited, it is not necessary to display each node's venue picture, since many matches are held at the same venue. For this type of match, if the venue picture has already existed, display the schedule information beyond the picture. Suppose users continue to click swimming node, there will be have a judgment by the system—if beyond the picture which the swimming node corresponding still have other schedule information, it would only delete the swimming node's information, or it would delete node's information and picture.

V. CONCULSION

As the treemap layout is limited to visualize multi-dimensional hierarchical Olympic Games data, we present a novel extension called magic treemap cube, which improves the layout and has a good performance on visualizing unfixed (cross-level) and complex data, inverting hierarchy without reorganizing the structure, and enabling users to easily compare among treemaps. For demonstrating the characteristics, an application (called OlympicVis) has been implemented to visualize the Beijing 2008 Olympic Games information, which involves multi-dimensional hierarchy and interconnected data. The OlympicVis can display Cube View, Schedule View, Medal View, Date Filter and Time Line.

ACKNOWLEDGEMENT

REFERENCES

[1] S G Eick. Graphically Displaying Text. *Journal of Computational and Graphical Statistics*. 1994, 3(2): 127-142.
[2] W S Cleveland. Visualizing Data. *Hobart Press*, 1993.
[3] G G Robertson, J D Mackinlay, S K Card. Cone Trees: animated 3D visualizations of hierarchical information. In: *ACM Conference on Human Factors in Computing Systems*. New Orleans, Louisiana, United States: ACM, 1991. 189-194.
[4] H Lu, J Fogarty. Cascaded treemaps: examining the visibility and stability of structure in treemaps. In: *ACM International Conference on Graphics Interface*. Windsor, Ontario, Canada: Canadian Information Processing Society, 2008. 259-266.
[5] B Johnson, B Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: *IEEE Conference*

on Visualization. San Diego, California, USA: IEEE Computer Society Press, 1991. 284-291.

[6]   B Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics (TOG).* 1992, 11(1): 92-99.

[7]   A Orso, J Jones, M J Harrold. Visualization of program-execution data for deployed software. In: *ACM Symposium on Software Visualization.* San Diego, California, USA: ACM Press, 2003. 67-76.

[8]   B B Bederson. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In: *ACM Symposium on User Interface Software and Technology.* Orlando, Florida, USA: ACM Press, 2001. 71-80.

[9]   A Fiore, M A Smith. Treemap Visualizations of Newsgroups. Technical Report, MSR-TR-2001-94, Microsoft Research, Microsoft Corporation: Redmond, WA, 2001

[10] M Smith. Tools for navigating large social cyberspaces. *Communications of the ACM.* 2002, 45(4): 51-55.

[11] J J Van Wijk, H Van de Wetering. Cushion treemaps: visualization of hierarchical information. In: *IEEE Symposium on Information Visualization (InfoVis).*San Francisco, CA, USA: 1999. 73-78, 147.

[12] F Van Ham, J J Van Wijk. Beamtrees: compact visualization of large hierarchies. In: *IEEE Symposium on Information Visualization (InfoVis).*2002. 93-100.

[13] P Demian, R Fruchter. Finding and understanding reusable designs from large hierarchical repositories. *Information Visualization.* 2006, 5(1): 28-46.

[14] F B Viegas, M Wattenberg, F Van Ham, et al. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics (TVCG).* 2007, 13(6): 1121-1128.

[15] J Stasko, E Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In: *IEEE Symposium on Information Visualization (InfoVis).*2000. 57-65.

[16] S K Card, J D Mackinlay, B Shneiderman. Readings in information visualization: using vision to think. *Morgan Kaufmann*, 1999.

[17] J Lamping, R,Rao P Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: *ACM Conference on Human Factors in Computing Systems.* Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co., 1995. 401-408

Xiaowu Chen received Ph.D. degree at Beihang University in 2001. He is a professor in School of Computer Science & Engineering and State Key Laboratory of Virtual Reality Technology & Systems, Beihang University. His research interests include computer graphics, visualization, virtual reality and augmented reality.

Haolin Yang is a master candidate in School of Computer Science & Engineering and State Key Laboratory of Virtual Reality Technology & Systems, Beihang University. His research interests include information visualization and video processing.

Yongtao Ma received master degree at Beihang University in 2010. His research interests include information visualization and information integration & retrieva

Bin Zhou is a Ph.D. candidate in School of Computer Science & Engineering and State Key Laboratory of Virtual Reality Technology & Systems, Beihang University. His research interests include computer vision, video processing and visualization.