An Efficient Visualization Technique of Large-Scale Nodes Structure with Linked Information

Su-Youl Mun*, Seok-Wun Ha**, *Member, KIMICS*

Abstract—This study is to suggest a visualization technique to display the relations of associated data in an optimal way when trying to display the whole data on a limited space by dealing with a large amount of data with linked information. For example, if you track an IP address through several steps and display the data on a screen, or if you visualize the human gene information on a 3-dimensional space, then it becomes even easier to understand the data flow in such cases.

In order to simulate the technique given in this study, the given algorithm was applied to a large number of nodes made in a random fashion to optimize the data and we visually observed the result. According to the result, the technique given in this study is more efficient than any previous method in terms of visualization and utilizing space and allows to more easily understand the whole structure of a node because it consists of sub-groups.

Index Terms—About four key words or phrases in Visualization, Nodemap, 2D, 3D Graphics

I. INTRODUCTION

As the performance of computer has advanced, now even a personal computer can handle a considerable amount of data and long-time-taking tasks. Also, the improvement of society came to allow a computer to solve problems that cannot be solved in the real world. The display of computer, however, does not meet the need yet.

This study is to suggest a visualization technique to display the relations of associated data in an optimal way when trying to display the whole data on a limited space by dealing with a large amount of data with linked information.

For example, if you track an IP address through several steps and display the data on a screen, or if you visualize the human gene information on a 3-dimensional space, then it becomes even easier to understand the data flow in such cases.

In order to simulate the technique given in this study, the given algorithm was applied to a large number of nodes made in a random fashion to optimize the data and we visually observed the result.

II. Related Studies

Studies related to visualization of data include the visualization of tree map[1][2][3], the use of fractal[4] and the fisheye lens algorithm[5].

The visualization of tree map is to display data on a 2-dimensional plane using a tree structure. Therefore, it is difficult to apply it to others than a tree structure.

The use of fractal is more flexible than the tree map. Its computation is, however, very complex, so it requires a long time.

Visualizing a large amount of data on a limited space, the fisheye lens algorithm expands and visualizes a specific area rather than the whole area. With this algorithm, you can take advantage of a limited space efficiently. However, if you want to display the whole data, it requires much of scrolling.

Recently, based on 3D technologies and built-in knowledge information, 3D systems have been developed so that a large data can be displayed in three dimensions and can be dynamically re-visualized after update[6][7].

Figure 1 shows the implementation of a node map using the formula (2-1) supplied in the tree map[1].

- The turning angle of a node within a group
  =The turning angle of a group/(The number of nodes within a group+1)
- The turning angle assigned to a node=360/The total number of nodes
- The turning angle of a group=The total number of nodes within a group*The turning angle assigned to a node (2-1)

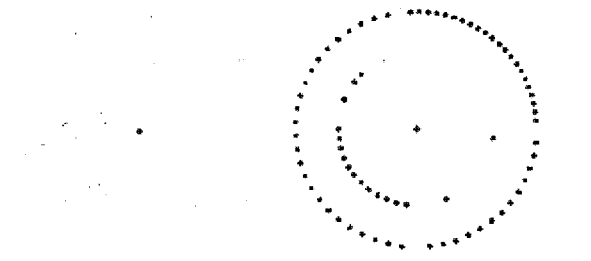


Fig. 1 The implementation of a node map with the tree map

This method has no problem in case of a small amount of data (Left), but a large amount of data (Right) makes linked data overlapped so that it becomes difficult to determine the relations between data and to understand the whole structure in a systematic way.

This method fits just a small amount of data. There

Manuscript received February 21, 2005.

*S. Y. Mun is with the Dept. of Office Information, Jinju Health College, Jinju, Gyeongnam, Korea 660-757.(e-mail:su10@chc.ac.kr)

**S. W. Ha is with the Dept. of Computer Science, Gyeongsang National Uninersity, Jinju, Gyeongnam, Korea 660-701. (e-mail:swha@nongae.gsnu.ac.kr)

has been no study on visualizing a large data on a limited area by saving the area itself.

In conclusion, this study suggests a visualization of a large data with linked information in a limited space. We try to optimize a limited area, using linked information and node data.

III. System Design and Implementation

A. Structure

The specification of the system suggested in this study is as in Figure 2.

- Hardware
  - -Model: IBM PC-Pentium IV
  - -Speed: 3.2 GHz
  - -Memory: 1 Gbyte
- OS: Windows XP Professional
- Language: JAVA
- Execution Environment: JDK 1.4.2

Fig. 2 System specification

The structure of system for simulation is as follows: First, nodes to be used in the simulation will be created. Such nodes with linked information will be then sorted on the center by a descending order for the optimization of space. Finally, the nodes will be arranged on the screen based on the linked information for weight. The result is shown as in Figure.

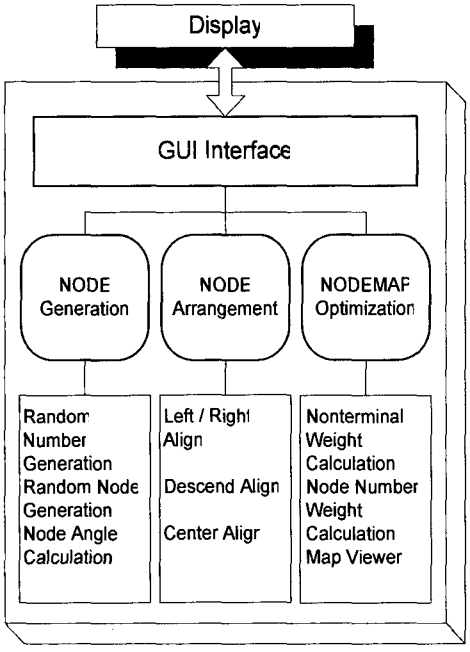


Fig. 3 The structure of system

B. System design

1. Node Format Design

This study is to optimize the space for a large data to be displayed and then visualize the data. Therefore, the system must be designed to visualize nodes on a given space without limit. You can find the format of random nodes in Figure.

1	1	3자리	3자리	3자리	3자리	3자리	3자리
그룹	레벨	현재 노드	부모 노드	자식 노드	자식 노드수	X좌표	Y좌표
3~5	3~5	000 ~999	000 ~999	000 ~999	000 ~999	000 ~999	000 ~999

Fig. 4 Node format

2. Deployment and Design of Node Map

In order to effectively display a large amount of data on a computer screen, a square space similar to a monitor is better than others. However, a circular space is more efficient than a square one with the same area[1].

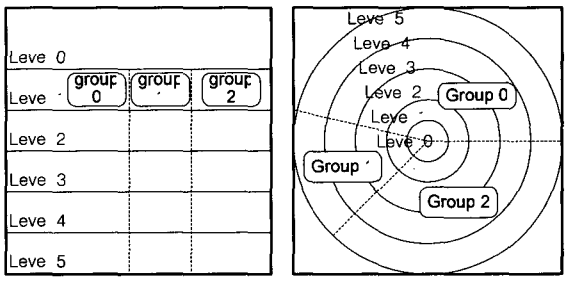


Fig. 5 Comparing space area between square and circle

In Figure 5 above, the left is the case that a space for each group is assigned in a square while the right is one in a circle. With the same area, the circle is more efficient than the square in displaying data.

Therefore, this study implements system with 2D circular space taking into consideration the characteristics of a large and random data.

3. Turning of Node

To display a node with linked information on a screen, the formula as in Figure 6 is used. The left figure in Figure 6 shows the movement from (x,y) to (x',y') with of turning angle given while the right one is the case that on object turns along with random points based on (XR,YR). is an angle between a horizontal line and the original point of coordinates for the initial point.

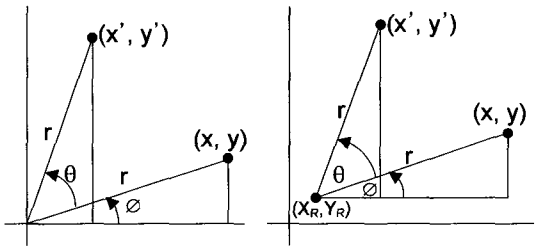


Fig. 6 Turning node on the basis of the original point and the base point (X_R,Y_R)

With the turning conversion formula for node, a formula (3-1) can be obtained between sides of a triangle and angle.

$$x' = r \cdot \cos(\phi+\theta) = r \cdot \cos\phi \cdot \cos\theta - r \cdot \sin\phi \cdot \sin\theta$$
$$y' = r \cdot \sin(\phi+\theta) = r \cdot \sin\phi \cdot \cos\theta + r \cdot \cos\phi \cdot \sin\theta \quad (3\text{-}')$$

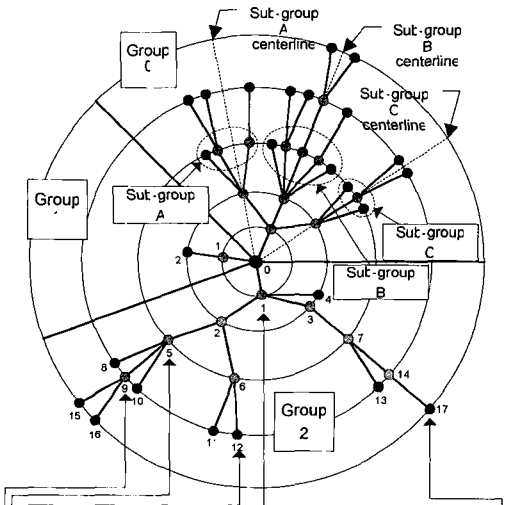
Therefore, this study displays nodes using a formula (3-2) based on a specific point on a screen but not the original point on coordinates.

$$x' = x_R - (x - x_R) \cdot \cos\theta - (y - y_R) \cdot \sin\theta$$
$$y' = y_R + (y - y_R) \cdot \cos\theta + (x - x_R) \cdot \sin\theta \quad (3\text{-}2)$$

4. Output Design

To display nodes created randomly on a screen, the system can be implemented without any restriction. However, the system will be designed to display up to 500 nodes taking the performance of test system into consideration. Also, the system will implement level 3 through 5 in depth and 3 through 5 groups. Figure 7 shows the position of nodes on a screen.

When designing an output screen, there are many ways to arrange the position of nodes. Here, nodes will be displayed by the unit of sub-group on the basis of the central point for each sub-group.



group	level	current node	parent node	child node	child count	X position	Y positior
2	4	012	006	000	000	367	343
2	3	005	002	008	003	296	247
2	4	009	005	015	002	256	287
2		001	000	002	003	212	218
2	5	017	014	999	000	376	335

Fig. 7 Output design of random nodes

In Figure 7, to display <sub-group A> on a screen, the central point of the sub-group is set and then it is displayed on the right and the left of the point. It is also true of <sub-group B> or <sub-group C>.

There are many ways to set the central point of a sub-group; here, the position of the parent node is used. This method will be able to more efficiently use the space within a group when applying the same radian value to both top levels of node and sub-nodes. However, a specific sub-group with more nodes than others might cause an overlap of nodes between sub-groups. This problem will be cleared using the weight of node and that of the central point of sub-group.

5. Selection of the Central Node in a Sub-group

The top level of node, or parent node, is used to obtain the central node to locate in the center of sub-group, shown as in Figure 9.In Figure 9, solid lines represent the position of the center in each sub-group from the central node in Level 2.

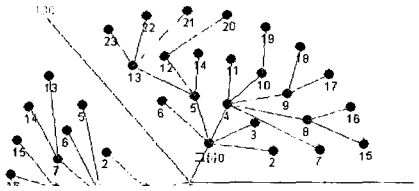


Fig. 8 Setting the center of a sub-group-1

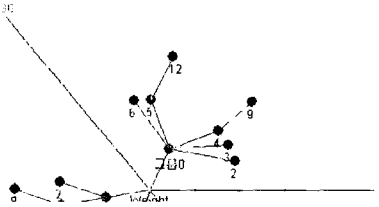


Fig. 9 Setting the center of a sub-group-2

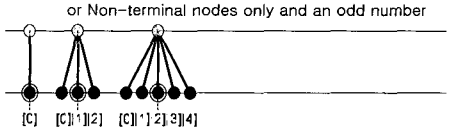
In spite of setting the central position of each sub-group, the center must be vary depending on the type and number of nodes in a sub-group. The reason is as follows.

In Figure 9, the 9th node in Group 0 is on the center of the sub-group while the 12th is out of the center. The reason is that the central position of a sub-group depends on the type and number of nodes contained in the sub-group. In Figure 8, the 5th sub-nodes in Group 0 include two non-terminal nodes and one terminal node.

The central position of a sub-group is affected not by terminal node but by non-terminal node. An even number of non-terminal nodes makes the central position out of the central line by weight, shown as in Figure 9.

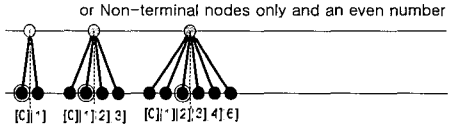
Once the center of a sub-group is set, it is important to find a node to locate in the center (◉ or ◎ marked). There are usually four cases that the selection of a node to locate in the center should be made. The following formulas give such a node in each case.

Case① – Terminal nodes only and an odd number
or Non-terminal nodes only and an odd number



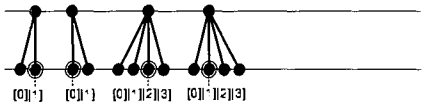
∴Node to locate in the center of a sub-group = Number of sub-groups / 2

Case②–Terminal nodesonly and an even number
or Non-terminal nodes only and an even number



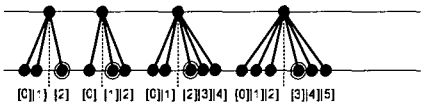
∴Node to locate in the center of a sub-group
= (Number of sub-groups / 2) –1

Case③–An odd number of non-terminal nodes
and an odd number of terminal nodes



∴Node to locate in the center of a sub-group =
· Central point as basis.
The right side has more terminal nodes than the left does
= Number of sub-groups / 2
· **Central point as basis.**
The left side has more terminal nodes than the right does
= Number of sub-groups / 2 - 1

Case④-An even number of non-terminal nodes and an odd number of terminal nodes



∴Node to locate in the center of a sub-group = Number of sub-groups / 2

As for Case ③, since if non-terminal nodes and terminal ones exist in a sub-group at the same time, only non-terminal nodes have linked information on their sub-nodes, terminal nodes cannot be the central node. That is why there are two formulas to select the central node.

For example, in case of five nodes in a sub-group and two non-terminal nodes among them, it belongs to Case ④ 5/2=2 (Remnants discarded). The node in array [2] becomes the central node of that sub-group.

C. System Implementation

1. Algorithm for Optimizing the Arrangement of Node Map

To efficiently arrange nodes randomly created on a limited space of a screen, first get nodes into line. The arrangement of nodes lined up on the center looks the shape of a fan so that a circular space can be more efficiently used.

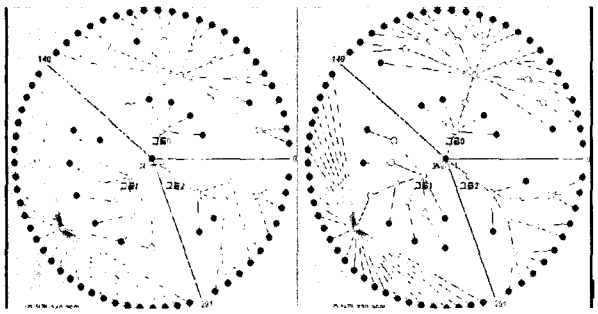


Fig. 10 Arrangement by the descending order (Left) and central arrangement (Right)

In Figure 10, the left is arrangement by the descending order of nodes randomly created while the right shows the central arrangement. As a result of this study, the central arrangement allows for a limited space to be more efficiently used than any left or right arrangement.

It is because a group space looks the shape of a fan similar to the structure of nodes with linked information. In addition, a side arrangement has the longer arc to display linked information that the central arrangement does, so that it is highly likely to cause the overlap of nodes and it is difficult to visualize the association among nodes.

In the central arrangement, sub-nodes must be lined up on the center on the basis of parent nodes. Or the overlap of linked information may take place. Therefore, arrangement must be preferably by depth from sub-nodes through parent nodes while output by width.

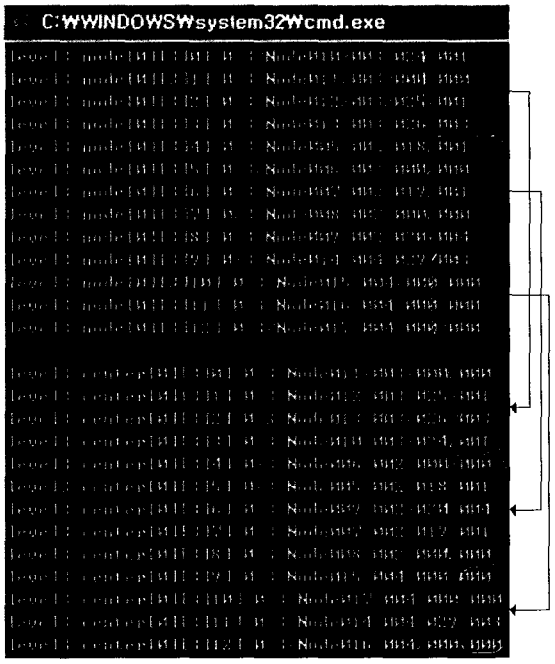


Fig. 11 The central arrangement of random node

Figure 11 shows the result of the central arrangement of nodes randomly created based on sub-group. The far right in a node represents the number of sub-nodes. You can see the sub-group specific central arrangement based on the number of sub-nodes.

2. Algorithm to Compute Weight for Optimization

When displaying the centrally arranged nodes on a screen, the overlap of links will be improved but there will need more improvement in the output of terminal nodes.

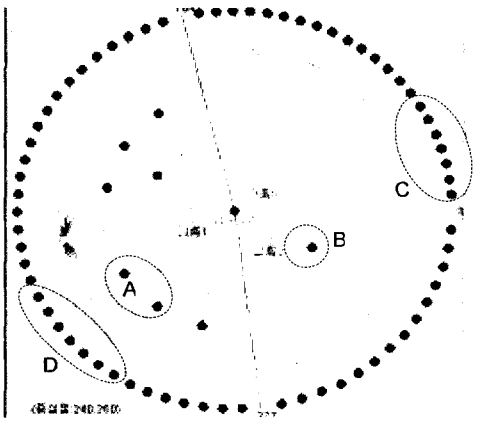


Fig. 12 Arrangement of terminal nodes

In Figure 12, terminal node A, B and C have no linked

information on sub-nodes but take up the same area as non-terminal nodes. It makes both the efficiency of space and the clarity of linked information of non-terminal nodes reduced.

<Step 1> Weight by whether having sub-nodes
① Weight for terminal nodes
- Default weight=0.3~0.5(Adjustable)

② Accumulated weight of terminal nodes
- (Node angle*Number of terminal)
  -(Node angle*Number of terminal*Weight for terminal)

③ Weight for non-terminal nodes
- Node angle+(Accumulated value for terminal nodes / Number of non-terminal nodes)

<Step 2> Weight for the number of sub-nodes
① Total weight for non-terminal
- Default weight for non-terminal*Number of non-terminal

② Weight for sub-nodes
- Total weight for non-terminal*(1/Number of sub-nodes)

③ Weight for the current node
- Number of sub-nodes*Weight for the sub-nodes

Fig. 13 Algorithm for computing weight

Indicating just whether there are terminal nodes and assigning the saved space to non-terminal nodes lead to even more optimized arrangement. In this study, the output of nodes is adjusted by assigning weight based on whether having sub-nodes and the number of sub-nodes.

Figure 13 shows the algorithm for computing weight. Figure 14 is to compare the positions of nodes before and after applying weight.

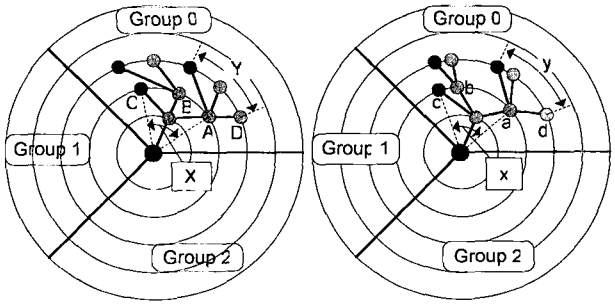


Fig. 14 The position of nodes before and after applying weight

- A Node Position=Node angle*1(First position)/Entire angle of Group 0
- B Node Position=Node angle*2(Second position)/Entire angle of Group 0
- C Node Position=Node angle*3(Third position)/Entire angle of Group 0
- a Node Position=Central angle of Group 0-(Weight for Node a/2)
- b Node Position=Central angle of Group 0+(Weight for Node b/2)
- c Node Position=Position of Node b+Weight for terminal node
- X area = x area
- Y area = y area (3-3)

Formula (3-3) shows change in the position of node before and after applying weight. The whole area of the sub-group is the same while nodes moved based on weight for an efficient arrangement. Figure 15 shows some part of the algorithm with Java.

```
public NonTerminalWeight
(double group_nodeangle, double group_angle,
 int level_count, int nonterminal_count,
 double terminal_weight){
  this.group_nodeangle = group_nodeangle;
  this.group_angle = group_angle;
  this.level_count = level_count;
  this.nonterminal_count = nonterminal_count;
  this.terminal_weight =  terminal_weight;}
public double nonterminal_angle(){ //
  terminal_count=level_count-nonterminal_count;//
  terminal_anglesum
  =group_nodeangle*terminal_weight*terminal_count;
   nonterminal_anglesum=group_angle-terminal_
   anglesum;
 if (nonterminal_count==0) {nonterminal_count=1; }
   nonterminal_weight
   =nonterminal_anglesum / nonterminal_count;
```

Fig. 15 Class to compute weight based on whether having terminal nodes

The central point in a sub-group uses the radian value of parent node regardless of sub-nodes. If any side has more nodes than the other, the overlap of sub-groups may be caused. Such overlap can be solved by adjusting the central points of sub-groups using weight. The total number of both terminal and non-terminal nodes is always consistent with the entire value for that sub-group so that the central point will not be out of the range of that group in any case.

IV. RESULT OF SIMULATION

A. Comparison with Studies Conducted Before

Figure 16 shows the comparison of the results by the tree map method [2] and this study.

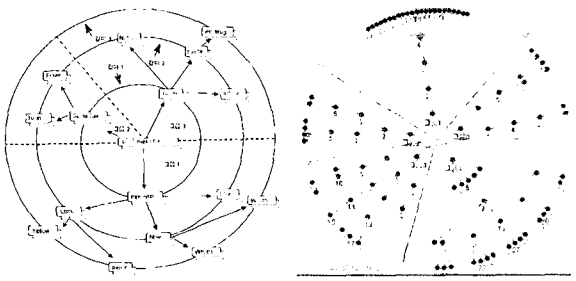


Fig. 16 Results of existing study (Left) and this study (Right)

The existing study implemented less than 20 pieces of data while this study displays a hundred pieces of data. Although this study displays far much data, it leaves much screen space in reserve than the existing study. Now, up to 500 nodes can be displayed.

Consequently, it can be said that the method suggested in this study is more efficient in terms of use of a limited space.

B. Comparison of Use of Space

This study can display much more data than the preceding study on the same space since this study use space more efficiently. The right figures in Figure 17 and 18 leave the greater space in reserve than the left ones. That means the space in reserve can be used for other purpose.

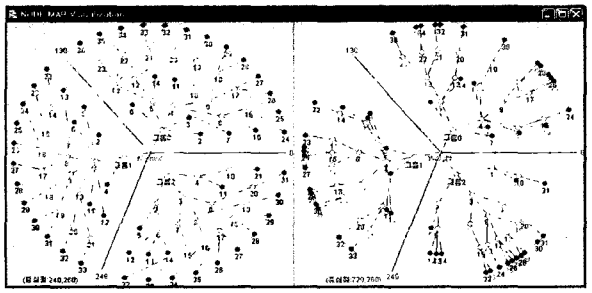


Fig. 17 Before (Left) and after (right) applying weigh-1

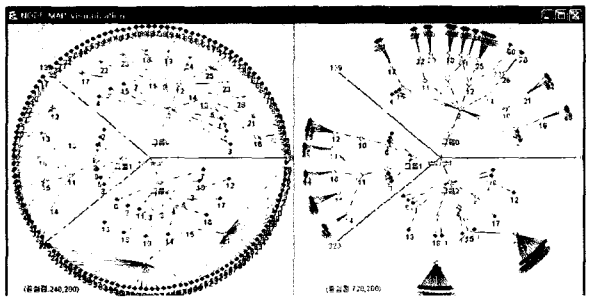


Fig. 18 Before (Left) and after (right) applying weigh-2

C. Comparison by Vision

Figure 19, 20, 21 and 22 shows a visual inspection of the result of increasing the number of nodes from 100 up to 500.

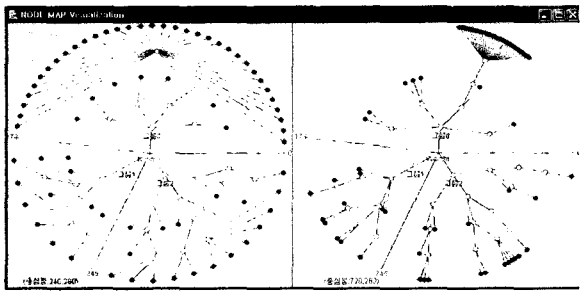


Fig. 19 Visual comparison 1 (100 nodes, 3 groups, levels)

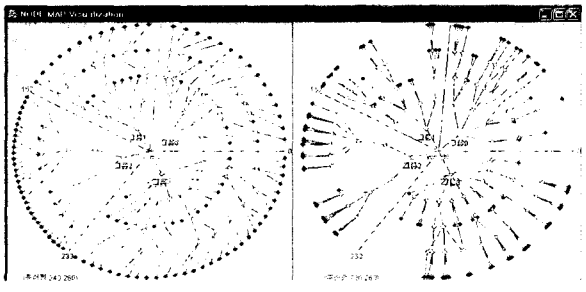


Fig. 20 Visual comparison 2 (200 nodes, 4 groups, levels)

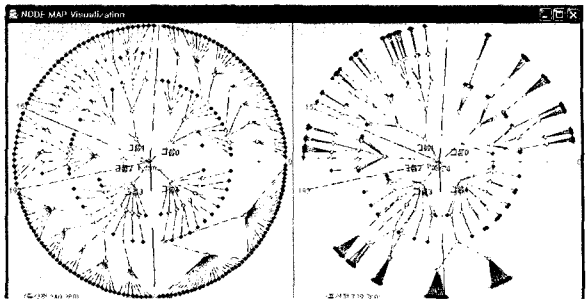


Fig. 21 Visual comparison 3 (300 nodes, 5 groups, 5 levels)

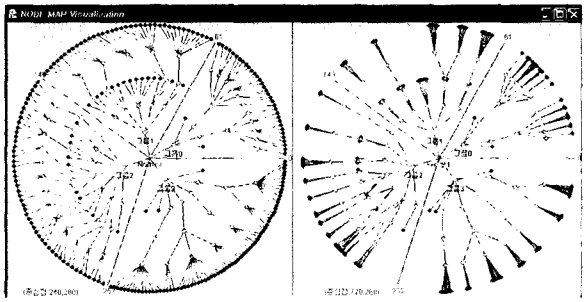


Fig. 22 Visual comparison 4 (300 nodes, 4 groups, 5 levels)

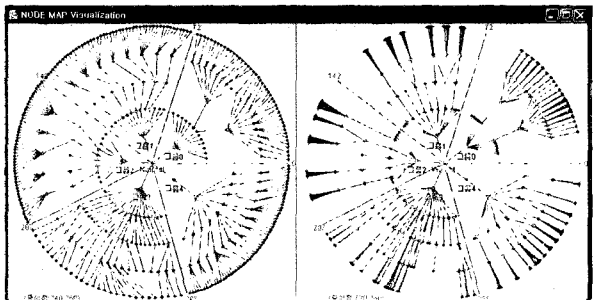


Fig. 23 Visual comparison 5 (400 nodes, 5 groups, 5 levels)

Here, the right figure applying weight uses space more efficiently than the left one, too. In addition, the right figure shows linked information between levels in a systematic way and the saved space can be used for other usage.

On a complex space such as in Figure 23, the overlap between sub-groups might take place since space for data to be displayed becomes reduced.

The reason is that information on sub-nodes cannot be known. It can be, however, solved by applying weight for the central line in that sub-group.

IV. CONCLUSIONS

This study suggests a visualization system using weight for the optimization of displaying a large amount of data with linked information on a limited space. It implements a visualization system to determine the structure of nodes with linked information on a limited screen space in a systematic way by applying weight

after the central arrangement of nodes randomly created.

Given the result of simulation, this study offers more efficient way to visualize data than any existing study. In addition, it is easier to figure out the entire structure of nodes due to the unit of sub-group.

This study makes it possible randomly create up to 500 nodes but the output could not be made due to the poor performance of system. In case of 500 nodes, the total of 1000 nodes including comparable ones must be displayed, so a system having higher specification is required. That matter should be dealt with by something more than PC.

For now, we must make a stronger system and implement a dynamic system to store data made preferably by width for each level on stack and correct it depending on the state of nodes in the next level for rearrangement.

REFERENCES

[1] Hee Youg Yoo, "AdvancedVisualization by an Efficient Information Layout and an Improved Fisheye Lens Algorithm", 동국대학교 석사학위 논문, p16, 1997.

[2] Jee-In Kim, "Visualization of Information in Hierarchical Structures", 건국기술연구논문지, 제 24 집, p 행 67, 1999.

[3] B.Johnson and B.Shneiderman, "Treemaps:a space-filling approach to the visualization of hierarchical information structures", In Visualization 1991, pages 284-291 IEEE, 1991.

[4] Hideki Koike and Hirotaka Yoshihara, "Fractal approaches for visualizing huge hierarchies", In Proceedings of the 1993 IEEE Symposium on Visual Lanuages. IEEE, 1993.

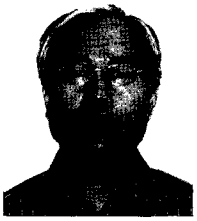
[5] Manojit Sarkar and Marc H. Brown, "Graphical fisheye views of graphs", In Proceedings of the ACM SIGCH Conference on Human Factors in Computing Systems, pages 83-91 ACM, April 1992.

[6] Andreas Nurnberger and Marcin Detyniecki, "Visualization Changes in Data Collections Using Growing Self-Organization Maps", IEEE, 2002.

[7] M.Balzer, A.Noack, O.Deussen and C.Lewerentz, "Software Landscapes:Visualizing the Structure of Large Software Systems", Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization, 2004.

Su-Youl Mun

Received the B.S. degrees in the Dept. of Computer science from Korea National Open University, Korea in 1989. and the M.S. degrees in Dept. of Educational Computer Science from Kyungnam University, Korea in 1991. He was finished doctoral course in Dept. Computer Science from Gyeongsang National University, Korea in 2004. Since 1998 to now, he has been an assistant professor, Dept. of Office Information, Jinju Health College, in Korea. His interests are image processing, multimedia, and data visualization.



Seok-Wun Ha

Received B.S., M.S., and Ph.D. degrees in Electronics Engineering from Pusan National University, Korea, in 1979, 1985, and 1995. He has been a visiting researcher in University of California, Riverside, in 2002. He is a professor in Dept. of Computer Science and a researcher at Research Institute of Computer Information & Communication and Engineering Research Institute in Gyeongsang National University. His major is computer vision and his interests are image processing, image retrieval, neural network, and data visualization.