Research Paper
Impact Factor: 4.164
Refereed Journal

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

HIERARCHICAL VISUALIZATION METHOD FOR MULTIDIMENSIONAL RELATIONAL DATA SET USING NR-PERFECT TREEMAPPING

K. Kalasha* **G. Nirmala****
**Head, PG and Research Department of Computer Science, Government Arts College, Thiruvannamalai.*
***Research Scholar, Government Arts College, Thiruvannamalai.*

Abstract
This paper describes multidimensional relational data sets visualization by using hierarchical method for enhanced treemapping. Many ideas behind by introducing a variety of interactive techniques for space optimization, rectangle overlapping and gaps and adjusting treemaps [1,2]. In this paper, we present strategies to visualize changes of hierarchical data using treemaps. A new NR-Perfect treemapping algorithm is presented to abrupt above all of these limitations. NR-Perfect treemapping algorithm would create rectangles with an aspect ratio close to one. The given size to form treemapping a rectangle can be formed in a big small split of a rectangle. Each rectangle followed by clearly few items of color, size, and position and represents a rectangle using graph based regions [3]. In this rectangle is cut out of a rectangle by substituting the values for T and it can be shown easily D the target aspect ratio is met. NR-Perfect treemapping well known treemap visualization in order to guarantee layouts with constant aspect ratio and it has effective power. When you implement this algorithm it satisfies many conditions and the implementation of NR-Perfect treemapping concept using python code.

Keywords: NR-Perfect (Node vs Rectangle) Treemapping, High Dimensional Data, Clustering, Multidimensional Relational Database.

I. Introduction
In this real world, to store and represent a huge amount of dataset into the database, we are in the need of multidimensional relational database. That is to be organized around a central theme and multiple views to see your huge amount of data sets. We can follow a millions of records that should be followed as high dimensional (for ex 1000 dimensions). “Data visualization is used to communicate data clearly and effectively through graphical representation [4].” It is an important technique to visualize multidimensional simultaneously.

A large amount of data set followed by high dimensionality would be difficult to visualize all dimensions at the same time. High dimensionality data will be visualize as “subset” (subspaces)[4]. Subspaces are partitioned into smaller subcategories. Subspaces are visualized in a hierarchical manner.

II. Hierarchical Visualization
“High dimensional data can be viewed using hierarchical method (“tree-structured”). Using small rectangular space to categories and organized each record”. This way exposes hierarchical visualization technique using **treemap** method [1]. Data visualization is closely related to computer graphics, multimedia system, human computer interaction, pattern recognition an etc. Making it a highly attractive and effective tool, the comprehension of data distribution and visualization techniques are used to discover data relationships. These have flexible mechanisms for mapping and to query result.

It was originally designed to support the interactive exploration of multidimensional relational data sets with rich hierarchical structure [5]. The dimensions within a data cube are often augmented with a hierarchical structure. This hierarchical structure may be derived from the semantic levels of detail within the dimension. The dimensionality of individual data elements can be compressed.

III. NR-Perfect Treemapping
Treemap defines a way to divide a rectangle into sub-rectangles of specified areas. It is a popular method for visualization hierarchical data [8]. By dividing the display area into rectangles recursively according to hierarchy structure the user selects data attribute [3]. Treemaps effectively display the overall hierarchy as-well-as the detailed attribute values from individual data entries. If a node is a leaf node, you can specify a size and color. If not, it will be displayed as a bounding box for leaf nodes.

Tree-map displays hierarchical data as a set of nested rectangles and information are categorized as unique in size and color. Within each category it partitioned into smaller subcategories. The rectangles are easy to visualize both size and color of the rectangle. It reflects the value of the color and rectangle to measure it. If data is in a hierarchy, a treemap is a good way to show all the values at once and keep the structure to visual. This is a quick way to make a treemap. Enhancing the treemap using its node and rectangle is called as NR-Perfect treemapping method.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Each node is displayed as a rectangle, sized and colored according to values that you are assigned in the treemap cell. This is useful when the cells are too small to customize. In Figure 1, the graph used to point the regions, that points starting and ending of the rectangle within the graph and many dimensions are used to visualize. Each dimension can be organized by each rectangle box that could be arranged to using coordinates points. ZOOM IN and ZOOM OUT are used to implement and manipulate the treemapping. ZOOM IN is nothing selected for some user views, ZOOM OUT means viewing particular dimension that could be process of showing, opening and viewing. These treemapping scroll pans are used to identify coordinates of each dimension. Graph based coordinates region set by user may be depend on size, user wish, highlighted, future use and etc.,

Merits
Easily analyzed by total cell, dimension and scrolled. Easily defined, treemapping visualized normally in the form of table but easily defined graph for understanding what it is? User based graph arranged by user not systematic, axis and units are arrangement based on user wish.

Each dimension compare to other dimension may be similar or may be not similar (consider as size, color, height, width). So that, why each dimension can be represented by using graph, that is graph units and axis may be changed with another dimension.

Example: d1 = 100kb, d2 =1 GB, d1 = user wish (1, 5), d2 = user wish (5, 5).

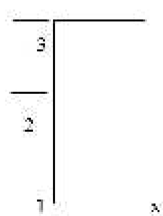

Figure 1: Graph based Rectangle Organization

In Table 1, the twelve categories of human disease and its sources are used for treemapping concept to visualize the data. In which few categories have subcategories that determined the nested rectangle. The subcategories are

Table 1: Human Disease Categories

S. No	Disease name	source of disease
1.	Actinomycosis	Gerenuseriae
2.	Afican sleeping sickness	Trypanosomabruei
3.	AIDS	HIV
4.	Bacerial vaginosis	Vaginosis microbiota
5.	Carrion’s disease	Bartonella bacilliform
6.	Toxocariasis	Toxocara canis
7.	Cholera	Vibrio cholera
8.	dengue fever	Flavi viruses
9.	Enterobiasis	Enterobius vermicular
10.	Sporotrichosis	Sporothrix schenckili
11.	Measles	Measles virus
12.	HFMD	Coxsackie a virus

Shown in Figure 2 and Figure 3. It is illustrated by space optimization on treemapping. The dimensions are no need to organize another rectangle to simplify the nested rectangle.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

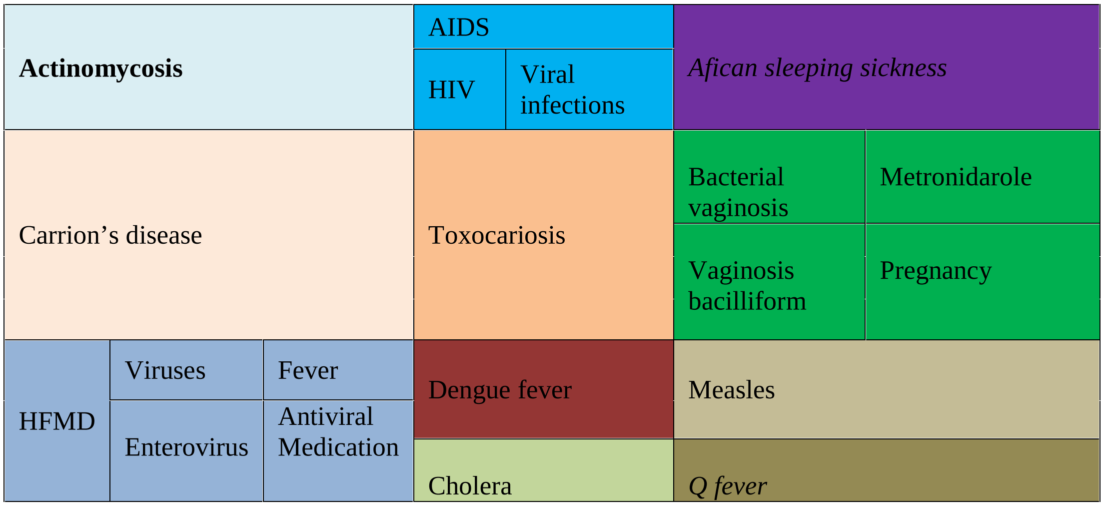

Figure 2: Treemapping of human disease categories

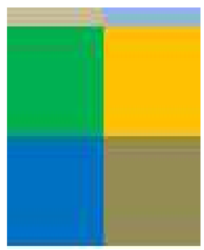

Figure 3: Nested Rectangles

IV. NR-Perfect (node vs rectangle) **Treemapping Algorithm**

NR-Perfect treemapping algorithm can be used if it satisfies the following norms

- **a)** Small snapshot are size based on some conditions (greater than of 0.000) and smaller rectangle must be followed intensity of pixel (aspect ratio) for need to select the rectangle for clear visualization (not blurred).
- **b)** Large rectangle size must be followed to organize nested rectangle. Figure 4, shows reduction of large rectangle to nested rectangles.

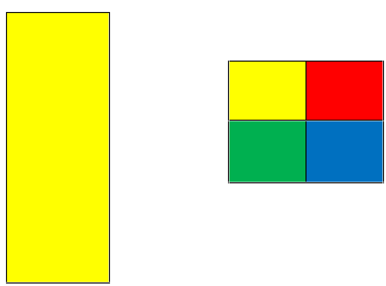

Figure 4: Large rectangle to small nested rectangles

- **c)** Each rectangle must have unique properties such as color, size, position and type of objects.
- **d)** Correlation between rectangles may be clustered. In figure 5, related data will be clustered in individual rectangle. It's one of the advantage of NR-Perfect treemapping, within this rectangle the followed sets of related (clustered) data can be exploded the number of rectangles.

Figure 5: Clustered Rectangles

e) Scaling between the rectangles: No gaps and overlapping are allowed. In figure 6, the human diseases categories are hierarchically represented by using 16 nodes are labeled by using treemapping.

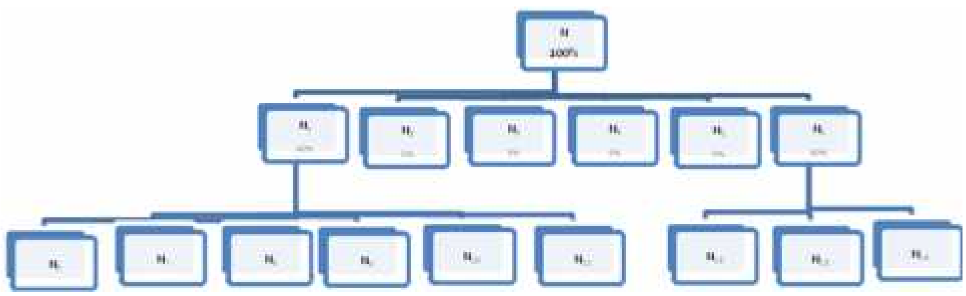

Figure 6: Hierarchical representation of human disease categories

Algorithm

N- dimensional data in dataset D project hierarchically defines the M- rectangles in treemap T, it must be less then (M<N) .

Input: N-dimensional relational dataset in D.

Output: project to M-rectangles in treemap T = {M_1, M_2, ……… M_n} // M_i > = M_{i+1}

1. Repeat Each dataset have specified the size that it depends the project with the rectangle size.
   // N_i 700kb, screen size1366px ×768px, viewport size 980×1280
   // M_i specified percentage (100%) of rectangle size
2. May be nested rectangle Mi → Mi1, Mi2, …………, Mik
3. Switch shape of S do region L1
4. Case Rectangle
5. if M1 > 2/3 then
6. Let T1 = {M1} and T2= T/{M1}
7. if M1 <= 1- AC/3 then // AC = aspect ratio
8. Cut S into two rectangle containers S1 and S2 with areas M (T1) and M(T2) // C = container
9. Else
10. Partition T into two subsets T1 and T2, such that ½ <=M(T1) <=2/3.
11. Cut S into two rectangle containers S1 and S2, with areas M(T1) and M(T2)
12. For nested data set will be proceed Mi+1(repeat statement 2)
13. Next data set will be proceed Ni+1(repeat statement 1)
14. ∑ Mi defines overall display information.
15. If new dimension F
    a) ADD F in N_i [N_i + F (2-dimension)]
    b) For loop each dimension in F = specified size

For example, human disease categories have 11 dimensional relational data set visualized simultaneously by implementing NR-perfect treemapping algorithm.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Here, view port is 980 ×1280 depend to structure data ,during implementation satisfied NR-perfect conditions, first taking first dimension from 11D [N = 11D] relational data set. contain content depend to structured percentage(may be 50%) of graph based rectangle sized for example 100 records should may be sized 4% of rectangle size(4×4)for M_1 at the same time coloring that rectangle using specified methods. Similar to structured for each dimension repetitively for second dimension have 2% of nested rectangle sized (2×2), third dimension have 5% of rectangle sized (5×5), fourth dimension have 4% of rectangle sized (4×4), fifth dimension have 4% of rectangle sized (4×4), sixth dimension have 2%3% of nested rectangle sized (2×2)(3×3), seventh dimension have 2%1% of nested rectangle sized (2×2)(1×1), eighth dimension have 3% of rectangle sized (3×3), ninth dimension have 2% of rectangle sized (2×2), tenth dimension have 4% of rectangle sized (4×4), eleventh dimension have 2% of rectangle sized (2×2).

If you newly added one dimension to human disease categories treemapping is not enlarge and overlapping viewport must followed as intensity of pixels adjacency of rectangles. For example, numbered 2, 6, 7, 8, 9. First one adjusting matrix of rectangle $\begin{bmatrix} 2\% & 2\% \\ & 2\% \end{bmatrix}$ followed as nested rectangles. Second one adjusting matrix of rectangle $\begin{bmatrix} 2\% & 3\% \\ 2\% & 3\% \end{bmatrix}$ followed as nested rectangles. Third one adjusting matrix of rectangle $\begin{bmatrix} 2\% & 1\% & 1\% \\ 0 & 1\% & 1\% \end{bmatrix}$ followed as nested rectangles.

V. Implementation of NR-Perfect Treemapping

PHYTHON language is used to implement the NR- Perfect treemapping to display simply rectangle division followed rectangle properties.

```
Import plotly. plotly as py
Import plotly. graph_objs as go
Import squarify
x = 0.
y = 0.
Width = 100.
Height = 100.

Values = [500, 400, 33, 25, 25, 17]

normed = squarify.normalize_sizes(values, width, height)
rects = squarify.squarify(normed, x, y, width, height)

# Choose colors
  color_brewer = ['rgb(166,206,227)','rgb(31,120,180)','rgb(178,223,138)',
         'rgb(51,160,44)','rgb(251,154,153)','rgb(227,26,28)']

Shapes = []
Annotations = []
Counter = 0

For r in rects:
    shapes.append(
        dict(
            type = 'rect',
            x0 = r['x'],
            y0 = r['y'],
            x1 = r['x']+r['dx'],
            y1 = r['y']+r['dy'],
            line = dict( width = 2 ),
            fillcolor = color_brewer[counter]
        )
    )
    annotations.append(
        dict(
            x = r['x']+(r['dx']/2),
```

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

```
            y = r['y']+(r['dy']/2),
            text = values[counter],
            showarrow = False
        )
    )
    counter = counter + 1
    if counter >= len(color_brewer):
        counter = 0

# For hover text
trace0 = go.Scatter(
    x = [ r['x']+(r['dx']/2) for r in rects ],
    y = [ r['y']+(r['dy']/2) for r in rects ],
    text = [ str(v) for v in values ],
    mode = 'text',
)

layout = dict(
    height=700,
    width=700,
    xaxis=dict(showgrid=False,zeroline=False),
    yaxis=dict(showgrid=False,zeroline=False),
    shapes=shapes,
    annotations=annotations,
    hovermode='closest'
)

# With hovertext
figure = dict(data=[trace0], layout=layout)

# Without hovertext
# figure = dict(data=[Scatter()], layout=layout)

py.iplot(figure, filename='squarify-treemap')
```

Output

Conclusion

We believe that, NR-Perfect treemapping approach helps to visualize multidimensional relational data set in a hierarchical manner. This algorithm satisfies the limitation of space optimization, overlapping rectangle gaps, and scaling. In this each rectangle justify the specified condition followed to exploded accurate positioned display area. Each rectangle have unique properties such as size, color and etc., these capabilities are utilized to organize relational dataset. This is an interactive visualization for improving the visibility of hierarchical Information structure. The treemap provides the hierarchical and categorical information in the 2D display in a 100% space-optimizing manner. We evaluate to display the space inefficiency of the traditional x-y axis bar chart, tree diagram, or a typical node-rectangle diagram. The space-optimizing characteristic is especially useful in the confines of a computer screen and under the trend towards data dashboards. We can even hypothesis that the treemap is more efficient in its display that reduces space utilization.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

References

1. C.Stotle, D.Tang, and P. Hanrahan. “Polaris: A system for query, analysis, and Visualization of Multidimensional relational database”. IEEE Transactions on visualization and computer graphics, Vol 1.8, No. 1, (pp.52-65) January 2002.
2. Shneiderman, B, “Tree Visualization with tree-maps: 2D Space filling approach ACM Transactions on graphics” (TOG), 11(1), 92-99, 1992.
3. A.Buja, D. Cook, and D.F. Swayne.”Interactive high dimensional data visualization”. In journal of computational and graphical statistics, 5(1), pp.78-99, 1996.
4. Johnson, B., & Shneiderman, B “Tree maps: A Space filling approach to the visualization of hierarchical information structures”. Visualization, 91 proceedings, IEEE conference on (pp. 284-291).IEEE in visualization, 1991.
5. S. Eick. “visualizing multi-dimensional data”. In computer graphics, pp, 61-77, February 2000.
6. Kong, N.Heer, J., & Agrawala, m. (2010). “Pereptual guideline for creating rectangle Tree-maps. Visualization and computer Graphics”, IEEE Transaction on, 16(6), 1990- 1998.
7. Sirin, E., & Yaman, F “Visualizing dynamic hierarchies in treemaps”, 2002.
8. J.Welling and M.Dertick. “Visualization of large multidimensional datasets”. In proc. Of the Future 2000.
9. J.wood and J.Dykes. “Spatially ordered treemaps”. IEEE Transactions on visualization and computer graphics, 14(6), 2008.
10. J.J.van wijk and h.van de wetering. Cushion treemaps:” visualization of hierarchical information In proc, IEEE symposium on information visualization”, pages 73-78, 1999.