Octal Graph Representation for Multi-Resolution 3D Landform Maps and Its Application

Goro Akagi * **Tomoe Motohashi** ** **Kenshi Nomaki** *** **Takeo Yaku** ****

ABSTRACT

We present a method of representing octal graph for rendering multi-resolution 3D landform maps. It is often necessary to transform landform data to change the resolution of specified areas, since fewer polygons is good when visualizing 3D landform. Landform data can effectively transformed by our method. For example, It is fine meshes in certain focused area, and somewhat rough meshes in other area by unification takes less time.

1. Introduction

3D landform maps are commonly used in visual communication. However, visualizing 3D landform maps with broad areas often requires an unfeasible amount of computation time, since they have a large number of polygons. Usually, one needs fine (50 m) meshes in certain focused areas and rather rough (e.q., 1 km) meshes in other areas. Such multi-resolution 3D maps may satisfy both a user's needs for fineness and rapid computational time. Thus, we will disscuss new methods of representing multi-resolution 3D landform maps. We recently proposed octet degree rectangular dissection graphs for general rectangular dissections in document processing [4]. We also applied these graphs to meshed map processing.
We modified and designed another octet degree rectangular graph for 3D maps. We propose list structure called H7-code in this paper. the H7-code satisfies both fineness and rapidity in the above sense, and corresponds to dynamic changes in maps.

2. Preliminaries

2.1.1 3D Landform Maps

3D landform maps (3dlms) are 3D Computer Graphic of landform (Fig. 1), and divides rectangles, whose nodes have attribute data (Fig. 2).

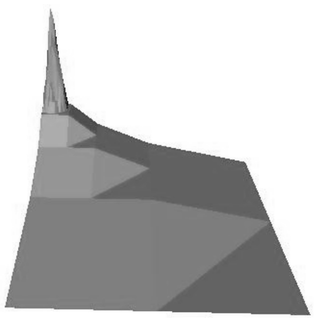


Figure1: 3D Landform maps

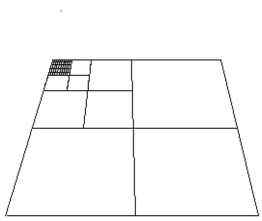


Figure2: rectangular

2.1.2 Rectangular dissection

In this paper, we deal with tables with heterogeneous rectangular dissections (compare Fig. 3 with 4).
Here, we review the definitions of tables and tabular diagrams given by Motohashi et al. [2].

Definition 2.1 [2] An (n,m)-*table* is a $\{(i,j) \mid 1 \le i \le n, 1 \le j \le m\}$ of integer pairs. A *table* is an (n,m)- table for some s and t. A *partial table* is a subset S of an (n,m) - table, where S is in the form of $\{(i,j) \mid k \le i \le l, s \le j \le t\}$ for integers $1 \le k$, $l \le n$, $1 \le s$, $t \le m$.

* Nihon University, akagi@ yaku.cs.chs.nihon-u.ac.jp

** Kanto Gakuin University, tomoe@kanto-gakuin.ac.jp

*** Nihon University, nomaki@ yaku.cs.chs.nihon-u.ac.jp

**** Nihon University, yaku@yaku.cs.chs.nihon-u.ac.jp

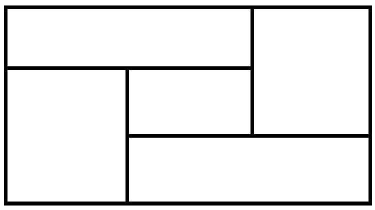


Fig. 3: heterogeneous rectangular dissection

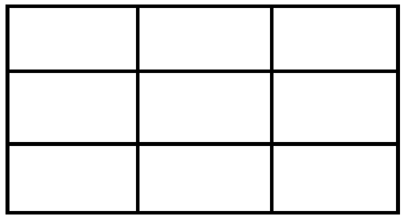


Fig. 4: homogeneous rectangular dissection

A partition P over a table T is a pair wise disjoint collection $S_1, S_2, ..., S_N$ of partial tables, where $S_1 \cup S_2 \cup ... \cup S_N = T$, and each S_i is called a *cell*. We call n *the row number* and m *the column number*.

Definition 2.2 [2] *The row grid* of an (n,m)- table T is a map $g_{row}(i)$: $\{0,1,...,s\} \rightarrow R$ such that $g_{row}(i) \leq g_{row}(i+1)$ for $0 \leq i \leq n-1$. *The column grid* is a map $g_{column}(j) : \{0,1,...,t\} \rightarrow R$ such that $g_{column}(j) \leq g_{column}(j+1)$ for $0 \leq i \leq m-1$. A grid is pair $g = (g_{row}, g_{column})$. *A tabular diagram* is a tuple $D = (T, P, g)$ of a table T, a partition P over T, and a grid g of T.

2.1.3 Tessellation Graphs ([2])

Let us also review the definition of attribute graphs representing tabular diagrams.

Definition 2.3 [2] *An attribute graph* is a 6-tuple $G = (V, E, L, \lambda, A, \alpha)$, where (V, E) is *a multi-edge undirected graph* with the set of *nodes* V and the set of *edges* E, L is the set of *labels* for edges, $\lambda : E \rightarrow L$ is *the label function*, A is the set of attributes, $\alpha : V \rightarrow A$ is *the attribute map*.

A tabular diagram $D = (T, P, g)$ is represented as an attribute graph $G_D = (V_D, E_D, L, \lambda_D, A_D, \alpha_D)$, where V_D is identified by a partition P (We denote a node corresponding to a cell c in P by V_c), E_D is defined by the following Rules 1-4, $L = \{enw, esw, eew, eww\}$, $\lambda_D : E_D \rightarrow L$ is defined by the following Rules 1-4, $A_D = R^4$ and $\alpha_D : V_D \rightarrow R^4$ are defined by the for by $\alpha_D(v_c) = (nw(c), sw(c), ew(c), ww(c))$.

In the following Rules 1-4, a label $\lambda_D(e)$ of e represents the relation between its starting and ending nodes.

Rule 1 [2] If $nw(c) = nw(d)$, i.e., c and d have the equal north wall, and there is no cell between c and d ,which have the equal north wall, then $[v_c, v_d]$ is in E_D and $\lambda_D [v_c, v_d] = enw$. In this case $[v_c, v_d]$ is called a north wall edge.

Rule 2 [2] If $sw(c) = sw(d)$, i.e., c and d have the equal south wall, and there is no cell between c and d which have the equal south wall, then $[v_c, v_d]$ is in E_D and $\lambda_D [v_c, v_d] = esw$. In this case $[v_c, v_d]$ is called a south wall edge.

Rule 3 [2] If $ew(c) = ew(d)$, i.e., c and d have the equal east wall, and there is no cell between c and d which have the equal east wall, then $[v_c, v_d]$ is in E_D and λ_D. In this case $[v_c, v_d]$ is called an east wall edge.

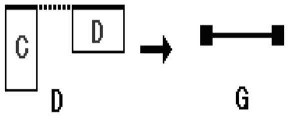


Fig. 5: Rule 1

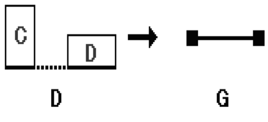


Fig. 6: Rule 2

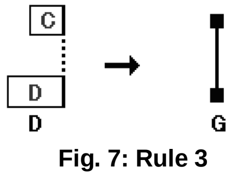


Fig. 7: Rule 3

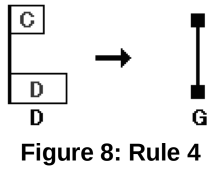
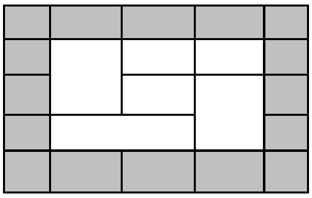


Figure 8: Rule 4

Rule 4 [2] If $ww(c) = ww(d)$, i.e., c and d have the equal west wall, and there is no cell between c and d which have the equal west wall, then $[v_c, v_d]$ is in E_D and $\lambda_D[v_c, v_d] = eww$. In this case $[v_c, v_d]$ is called a west wall edge.

2.2 H3-code

H3-code is a data structure that achieves the representation of tables based on tessellation graphs, and has been developed for the application to heterogeneous table processing in documents.



```
1 0 0
1 0 0 0 2 2 0 0 6 6 0 0 0 0 0 0 0
2 0 1 1 3 3 0 0 7 7 0 0 0 1 0 0 0
3 0 2 2 4 4 0 0 8 11 0 0 1 2 0 0 0
4 0 3 3 5 5 0 0 12 8 0 0 2 3 0 0 0
5 0 4 4 0 0 0 0 9 9 0 0 3 3 0 0 0
6 0 0 0 7 8 1 1 10 10 0 1 0 0 0 0 0
7 1 6 10 2 8 11 2 2 15 18 0 2 0 1 0 0 0
8 1 7 6 9 9 3 4 11 12 0 1 1 3 0 0 0
9 0 8 8 0 0 5 5 13 13 0 1 3 3 0 0 0
10 0 0 0 11 7 2 2 15 18 1 2 0 0 0 0 0
11 1 10 7 12 13 4 8 19 15 1 2 1 2 0 0 0
12 1 11 15 13 16 4 8 20 20 1 3 2 3 0 0 0
13 0 12 11 0 0 9 9 16 16 1 2 3 3 0 0 0
14 0 0 0 15 15 10 10 17 17 2 3 0 0 0 0 0
15 1 14 14 16 12 7 11 18 19 2 3 0 2 0 0 0
16 0 15 12 0 0 13 13 21 21 2 3 3 3 0 0 0
17 0 0 0 18 18 14 14 0 0 3 3 0 0 0 0 0
18 0 17 17 19 19 15 7 0 0 3 3 0 1 0 0 0
19 0 18 18 20 20 11 15 0 0 3 3 1 2 0 0 0
20 0 19 19 21 21 12 12 0 0 3 3 2 3 0 0 0
21 0 20 20 0 0 16 16 0 0 3 3 3 3 0 0 0
```

Fig. 9: Sample of heterogeneous table and corresponding H3-code

2.3 Operations on Grid Graph

Unify-Cells unifies two adjacent inner cells in a rectangular dissection([3]).

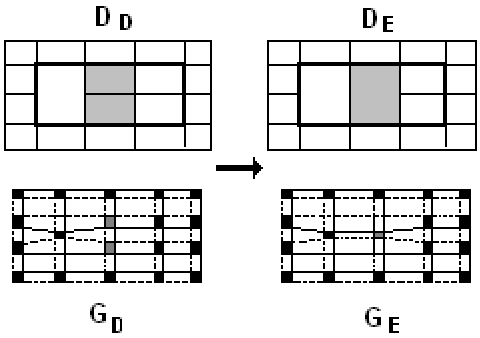


Fig. 10: Changes in table and corresponding graph via Unify-Cells

3. Landform Octal Graph

Definition 3.1 *Landform Octal Graph* is an attribute graph $G_D = (V_D, E_D, L, \lambda_D, A_D, \alpha_D)$ which represents a tabular diagram $D = (T, P, g)$, where

V_D is identified by a partition P (We denote a node corresponding to a cell c in P is denoted by V_c), E_D is defined by the following Rules 1-4, $L = \{enw, esw, eew, eww\}$, $\lambda_D : E_D \to L$ is defined by the following Rules 1-4, $A_D = R^4$ and $\alpha_D : V_D \to R^{32}$ are defined by $\alpha_D(v_c) = (nw(c), sw(c), ew(c), ww(c), att_{1.}, ..., att_{22})$.

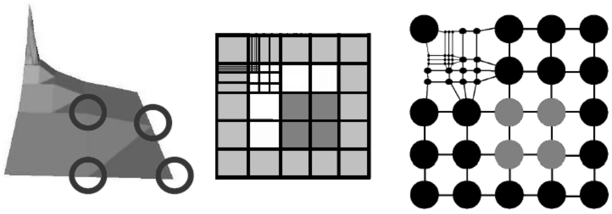


Figure 11: 3D landform maps and corresponding rectangular dissection and landform octal graph. The altitude at these red circle on this 3dlm are stored in these cells.

3.1 Characteristics of Landform Octal Graph

By using by octal graph operations for Unify Cell of Focused area to reduce polygons can be carried out simply in fewer minute. We show the differences of computation time for unify cell (fig.12). In the case of Rectangular dual, the computation time is proportional to time n. But, in the case of octal graph, the compution time does not depends on n,

	Unification of two adjacent cells
Rectangular Dual	O(n) n: number of cells
octal degree rectangular dissection graph	O(1)

Figure 12: The differences of computation time for unify cell

3.2 Example of landform octal graph

We show Example of landform octal graph(fig.13). The picture on left is 3D landform Map.

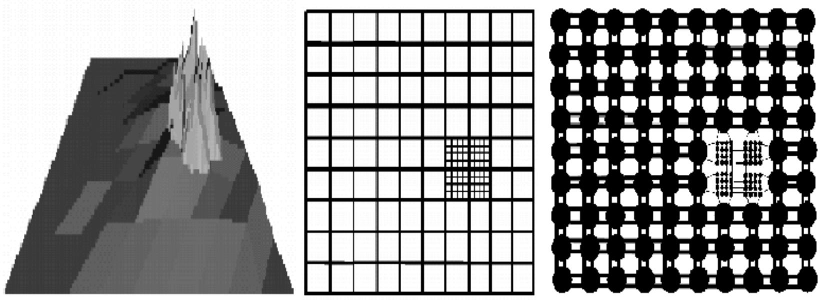


figure 13 : 3D landform Map(left). And corresponding Rectangular dissection (center) and landform octal .graph (right).

4. H7-code

Local splitting and unification of meshes are indispensable in generating 3D maps from numerical maps. Furthermore, the addition of geographical information to data is also required. The aim of this section is to present a new data structure and operations on such data to enable us to transform 3D maps (e.g., mesh unifications) and extract geographical features (e.g., ridges, valley, contour line) more effectively.

4.1 Outline of H7-code

Note that the construction of H7-code consists of four blocks.

(1) Header Block

The header block has basic information on rectangular dissection and consists of three fields (expect spare).

(2) List Block

The list block has information on the attributes of cells. The number of cells equals the number of records. This is 32 fields per a record(Fig. 14).

(3) Content Block

The content block has information on the picture to rapping 3D landform maps.

(4) Tabular layer block

The tabular layer block has information to manage rectangular dissection.

For example, the table in Fig. 15 is encoded with H7-code(Fig. 16). This type of data structure for the arrangement of cells was first introduced in H3-code [4]. The contents part involves data related to cells such as numbers and characters, which express altitudes, place-names, and pictures mounted to the surfaces of 3D maps. A picture of Mt. Fuji in Fig. 17 is produced form a H7-code file in Fig. 16.

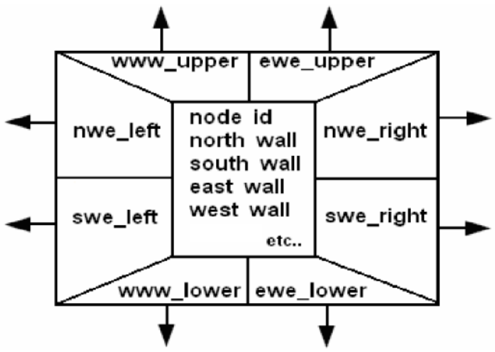


Fig. 14: detail of list block

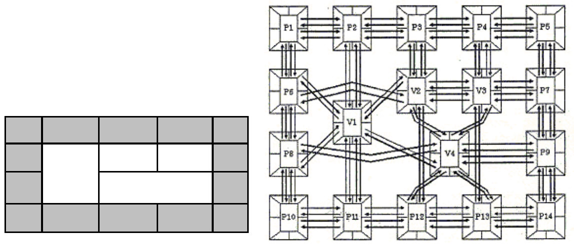


Fig. 15: Example table

```
#       This file was transformed by map2h7-50m Version 1.14
#               Date: 2005.08.31        Time: 17:13
#               MESH SIZE: 50m
#               MAP NAME:Mt. FUJI
#               MESH CODE: 533805
#               LINE:     001 - 200     (ORG: 001 - 200)
#               COLUMN:    001 - 200  (ORG: 001 - 200)

1.14 200 200 0 0 0 0 0
1 0 0 0 2 0 0 203 0 203 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
2 0 3 1 3 1 0 204 0 204 0 1 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
3 0 4 2 4 2 0 205 0 205 0 1 1 2 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
4 0 5 3 5 3 0 206 0 206 0 1 2 3 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
5 0 6 4 6 4 0 207 0 207 0 1 3 4 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
6 0 7 5 7 5 0 208 0 208 0 1 4 5 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
7 0 8 6 8 6 0 209 0 209 0 1 5 6 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
8 0 9 7 9 7 0 210 0 210 0 1 6 7 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
9 0 10 8 10 8 0 211 0 211 0 1 7 8 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
10 0 11 9 11 9 0 212 0 212 0 1 8 9 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
11 0 12 10 12 10 0 213 0 213 0 1 9 10 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
12 0 13 11 13 11 0 214 0 214 0 1 10 11 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
13 0 14 12 14 12 0 215 0 215 0 1 11 12 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
14 0 15 13 15 13 0 216 0 216 0 1 12 13 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
15 0 16 14 16 14 0 217 0 217 0 1 13 14 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
16 0 17 15 17 15 0 218 0 218 0 1 14 15 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
17 0 18 16 18 16 0 219 0 219 0 1 15 16 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
18 0 19 17 19 17 0 220 0 220 0 1 16 17 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
19 0 20 18 20 18 0 221 0 221 0 1 17 18 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
20 0 21 19 21 19 0 222 0 222 0 1 18 19 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0
                                :
```

Fig. 16: Sample H7-code files

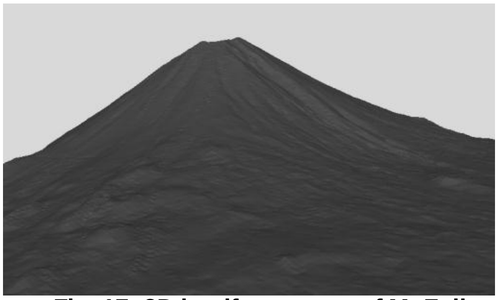


Fig. 17: 3D landform maps of Mt.Fuji

5. Application

We also report a conceptual model of a processing system based on H7-code, which we are developing for 3D map visualization. The concept for the visualization system based on H7-code is shown in Fig. 18.

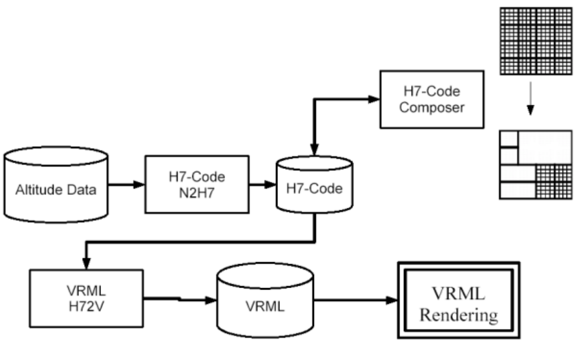


Fig. 18: Concept for visualization system based on H7-code

6. Comparison

Let us now discuss the advantages of our new data structure by comparing the number of meshes reduced with quadtree with those with landform octal graph.

We compared the landform octal graph with quadtree to confirm the advantages of the landform octal graph. We made a rectangular dissection of Nagano's altitude data, and confirmed the Unify-Cell of several methods.

The number of cell of original rectangular dissection is 64(Fig. 19). The number of cell of rectangular dissection to be made method of quad-tree is 55(Fig. 20). The number of cell of rectangular dissection to be made method of landform octal graph is 27(Fig. 21). Therefore we show to reduce number of cell by method of landform octal graph than quadtree.

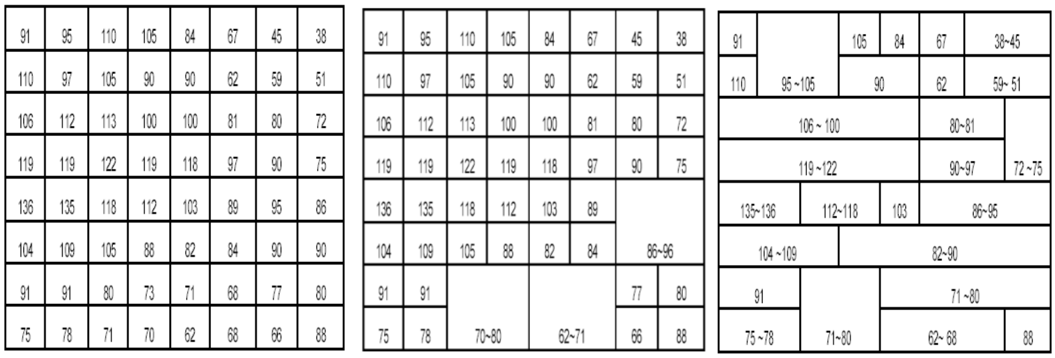


Fig. 19: Original rectangular dissection (left) and Quad-tree to be made method of landform octal graph (center) and Rectangular dissection to be made method of landform octal graph (right).

7. Conclusion

We proposed octal graph representation for multi-resolution 3D landform maps and a landform octal graph is composed of an octal graph, altitude, and color. We implemented a landform octal graph with H7-code and discussed the advantages of the landform octal graph over quadtree. We proposed a structure for a 3D landform map system and discussed the advantages of the landform octal graph. As for Future work, we implement a structure for a 3D landform map system and plan to achieve hexadecimal graph representation for stratum maps.

Acknowledgements
The authors wish to thank Youzou Miyadera and Kensei Tsuchida for fruitful discussion and helpful comments.

References
[1] J. L. Bentley, Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM 18(1975), 509-517.
[2] T. Motohashi, K. Tsuchida and T. Yaku, Attribute Graphs for Table and Their Algorithms, Proc, Foundation of Software Engineering 2002, K. Inoue Ed, Kindaikagakusha, 2003, 183-186.
[3] T. Kirishima, T. Motohashi, K. Tsuchida and T. Yaku, Table Processing Based on Attribute Graphs, Proc. 6th IASTED Conf. SEA(2004), 317-320.
[4] T. Arita, T.Motohashi, K. Tsuchida and T. Yaku, An Octet Degree Graph Representation for the Rectangular Dissections, 応用数学合同研究集会報告集, 2004,131-136.
[5] http://cai.chs.nihon-u.ac.jp/rr/cai-rr-05001/h7top.html