Flow direction algorithms in a Hierarchical Hexagonal Surface Model

Joseph W Wright, Antoni Moore & Greg Leonard

School of Surveying
University of Otago. Dunedin, New Zealand
Phone: +64 3 479-7585 Fax: +64 3 479-7586
Email: joseph.w.wright@otago.ac.nz

Presented at SIRC NZ 2013 – GIS and Remote Sensing Research Conference
University of Otago, Dunedin, New Zealand
August 29th – 30th 2013

***Keywords and phrases*:** Hexagonal Image Processing, Digital Elevation Model, Flow Direction, Level of Detail, Hierarchical Surface Model.

1.0 INTRODUCTION

This study proposes a Hierarchical Hexagonal Surface Model (HHSM) for storing and indexing large spatial arrays. HHSM uses an intrinsically hierarchical indexing system that creates interesting opportunities for neighbourhood based operations, multi-level analysis and forming adaptive surfaces. Research into surface models that depart from traditional uniform rectangular raster approaches is motivated by changes in data availability. Models developed for hydrological analysis of large areas make assumptions about geomorphology that are not always appropriate in urban areas, primarily due to the land shaping forces of human activity. The hydrological response of an urban surface compared to an undeveloped or rural area can vary signficantly due to important hydrological features that occur at very fine spatial scales. The regular spacing and inconsistent connectivity of conventional raster based surface models limit the capability of hydrological models to represent urban areas. Research has previously demonstrated hexagonal data sampling has advantages over rectangular grids for hydrological modelling (Brimkov et al, 2001, de Sousa et al, 2006). Therefore, hierarchical hexagonal data structures are needed that can adapt to the urban hydrological environment.

2.0 HHSM DATA STRUCTURE

2.1 Indexing

The proposed data structure uses the Hexagonal Image Processing (HIP) referencing system described in Middleton & Sivaswamy (2001). This referencing system shares many characteristic with the Quadkey used in Bing Maps Tile System (Microsoft, 2013). Characteristics in common include describing position with a single ordinate and having each position ordinate beginning with the ordinate of the pixel of the coarser Level of Detail (LOD) that contains it. Unlike the Quadkey, however, HIP uses a hexagonal partition of space. Each coarser HIP LOD is formed by aggregating 7 hexagons together in a Generalised Balanced Ternary (GBT) (Gibson & Lucas 1982). The value of each digit within a HIP index ranges from 0 to 6 and represents the rotation of a base vector by multiples of $\pi/3$ radians, except 0, which defines the centre of the GBT. The centre of each HIP cell is the centre of the 0 hexagon of its children. The outlines of the cells are only true hexagons on a single predefined base level (L0) and the orientation of the array rotates with each subsequent scaling. A three LOD HIP structure with both HIP ordinates and (x, y) coordinates is shown in Figure 1.

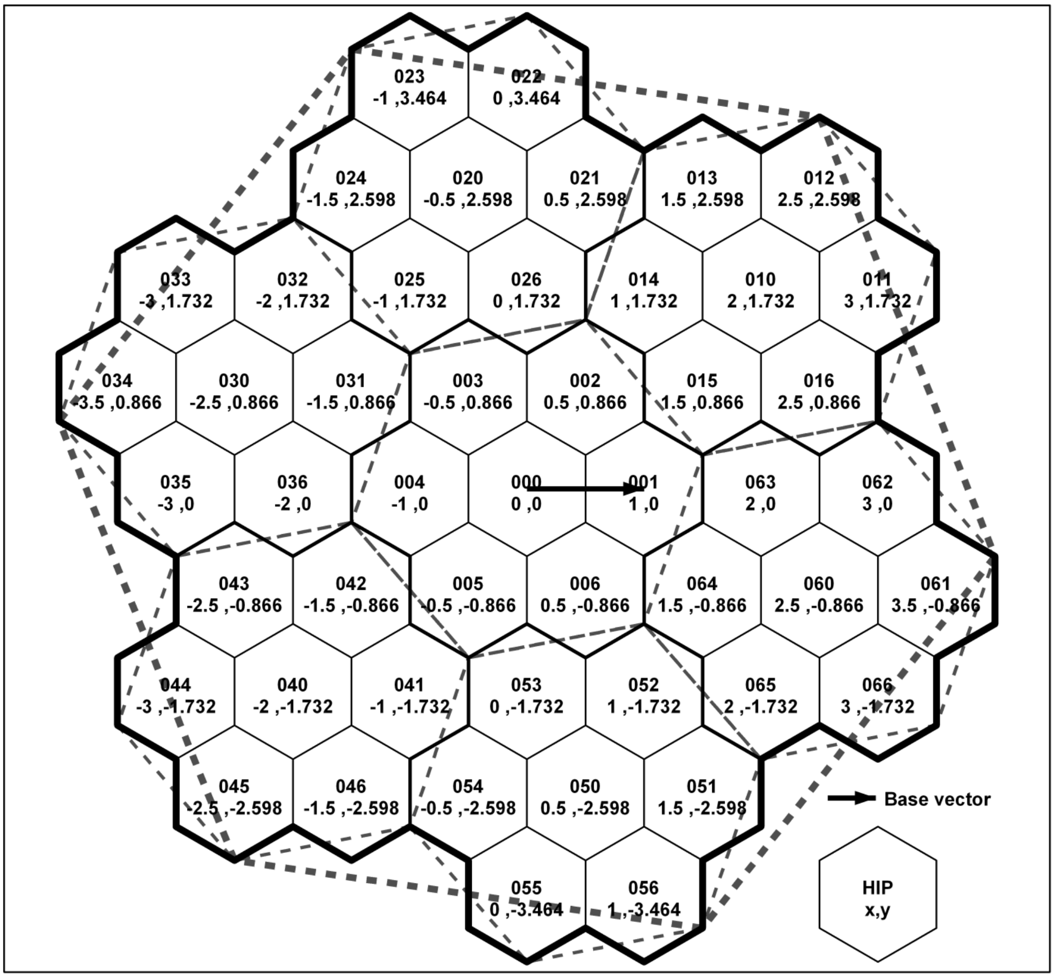

Figure 1: The HIP referencing system for part of a 3 level HIP partition of space. Hexagons are grouped into aggregates of 7 for each coarser LOD. Voronoi tessellations are shown for all three levels.

2.2 Storage

The HHSM decomposes the HIP array into consistently indexed tree and array components. The division between the two components is defined by an aggregation value that sets the maximum size for the array component. Data is stored as one or more arrays where each digit of the HIP index up to the aggregation value is the ordinate of a dimension within the array. As there are 7 possible locations within each hexagon, the array is a $7_1 \times 7_2 \cdots \times 7_n$ array, where n is the LOD. Therefore, the array part of a HIP ordinate is also the Cartesian coordinate of the value within the 7^n array. The remaining digits of the HIP ordinate define the key of a B-tree index on which the array is stored. Excessively large trees or arrays both adversely affect performance. To improve performance a combination of tree and array storage, implemented using the Python module PyTables, is used in this study. The combined tree-array format in PyTables can process arrays that are larger than can be stored in a system's memory.

2.3 Level of Detail Pyramids

LOD models permit representations of the same data at different resolutions. Frequently LOD can also vary across a terrain (Floriani et al 2005). They are typically formed by sub-sampling the data or by selective refinement and are often used to create multi-resolution models that vary in detail depending on distance from a view point, an application that requires real time rendering. Here the use of LOD is not to facilitate graphical display but to reduce the number of cells required for modelling without reducing model accuracy. A coarser LOD HHSM pyramid layer consists of a new array with 1/7th as many hexagons. HHSM can efficiently form LOD pyramids for two reasons. Firstly, the coarser resolution pixel HIP ordinate is the same as the beginning of its children in the finer levels of detail, making it a trivial task to relate the positions of cells in different LODs to each other. Secondly, the 7 values of a GBT at the finest LOD in an n level HIP are stored adjacent in dimension n of the array. This can be exploited to quickly summarize each hexagon, and in this fashion a pyramid of coarser resolutions can rapidly be built. A schematic of a 5-level data structure, with an aggregation value of 3 and its pyramid layers is shown in Figure 2. The coarser resolution aggregates are, in fact, not hexagons but complex fractal shapes. However, they can be simplified to the Voronoi tessellations of the centre of each cell, forming rotated hexagons (see Figure 1).

2.4 Adaptive Surfaces

An adaptive surface can be created by applying a decision rule to determine the LOD required for a given area. For instance, the decision rule may be that if all the L0 children of a given polygon are within a set limit of their mean, then only the parent is required in the adaptive surface. Applying such a rule is similar to a quad-tree division of space, effectively forming a hept-tree. It is possible to apply the hept-tree adaption algorithm to either elevation or flow direction arrays. The decision rule can be used to preserve hydrological significance minimising cell numbers without losing accuracy in representing hydrological behaviour.

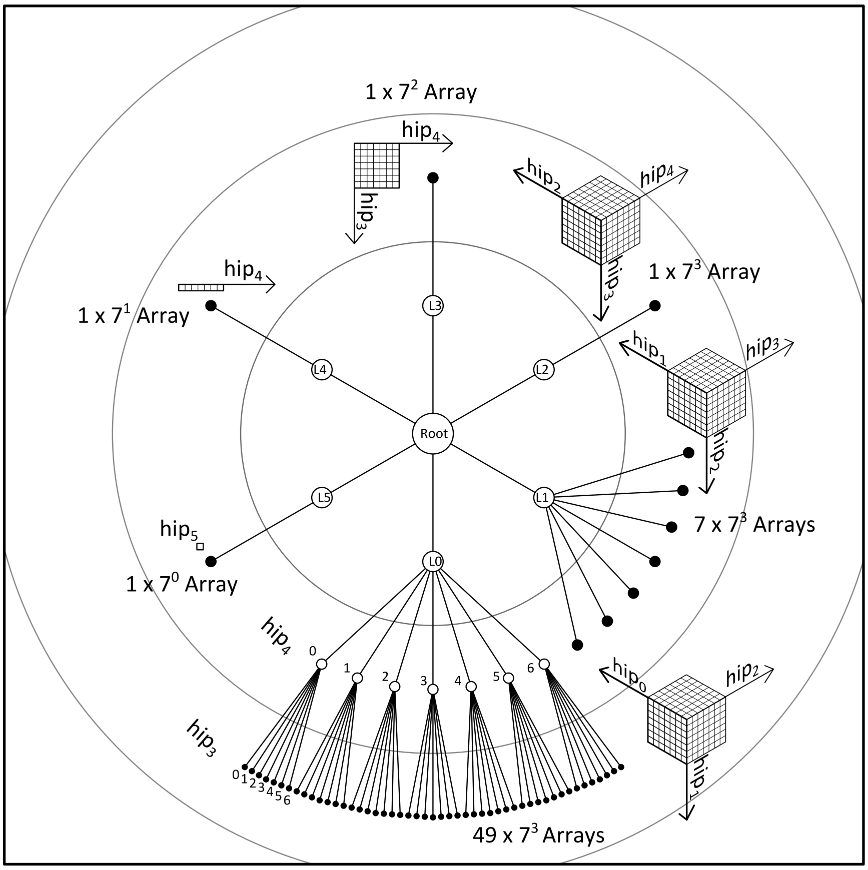

Figure 2: Schematic representation of the data structure of 5-level HHSM (L0) with pyramids (L1 – L5), aggregating at level 3. The 5-digit HIP index is decomposed to hip_4 hip_3 hip_2 hip_1 hip_0. The location of each digit is shown.

3.0 FLOW DIRECTION IN HHSM

There are a number of ways of determining D8 like flow direction solutions within the HHSM, which generate distinctive behaviours not seen in rectangular arrays. The basic premise of D8 flow direction is that flow is attributed in the direction of the steepest downslope neighbour (O'Callaghan and Mark, 1984). Many alternatives to this approach are possible, however, here only the D8 method is considered, with the additional factors of hexagonal sampling and adaptive cell size. The obvious corollory of the D8 in hexagonally sampled space is the D6, where the flow direction is the steepest downslope neighbour of the six nearest neighbours. However, for multi-level analysis and modelling, the D6 flow direction algorithm is complicated by the angle of rotation between HIP levels because the directions of the neighbours at a given level are not represented in the neighbourhoods of coarser and finer levels.

4.0 CONCLUSIONS

The HHSM is a robust and simple method to store and process hexagonally sampled data. A combination of tree and array storage permits very large arrays to be processed effectively. LOD pyramids, variable density surfaces and multi-level analysis are all possible using HHSM. Flow direction algorithms in HHSM have aspects not present in flat rectangular raster structures, which have potential to model complex surfaces more effectively.

ACKNOWLEDGEMENTS

Data for this project was provided by the Dunedin City Council.

REFERENCES

Brimkov V. and R. Barneva (2001) "Honeycomb" vs Square and Cubic Models, *Electronic Notes in Theoretical Computer Science*, 46, pp. 321-338.

de Sousa L, F Nery, R Sousa and J Matos (2006) Assessing the accuracy of hexagonal versus square tiled grids in preserving DEM surface flow directions, in Caetano MPM (Ed.), *The 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Accuracy 2006)*, Lisbon, pp. 191-200.

Floriani L. L. Kobbelt and E. Puppo (2005) A Survey on Data Structures for Level-of-Detail Models, in Dodgson, N. M. Floater, S Michael. and M Sabin (Ed.), *Mathematics and Visualization*, Springer, Berlin Heidelberg, pp. 49-74

Gibson L and D Lucas (1982) Spatial data processing using generalized balanced ternary, *Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing*, Las Vegas, Nevada, 14-17 June, pp. 14-17.

Middleton, L. and J. Sivaswamy (2001) Edge detection in a hexagonal-image processing framework. *Image and Vision Computing*, 19:14, pp. 1071-1081.

Microsoft (2013) Bing Maps Tile System. Accessed 20 June, 2013. <http://msdn.microsoft.com/en-us/library/bb259689.aspx>

O'Callaghan, J. and D. Mark (1984) The extraction of drainage networks from digital elevation data, *Computer Vision, Graphics, and Image Processing*, 3:28, pp. 323-344.