The Regular Hierarchical Surface Model (RHSM)

Joseph W. Wright[1 2], Antoni B. Moore[* 1] and Gregory H. Leonard[1]

[1]School of Surveying, University of Otago, PO Box 56, Dunedin, New Zealand
[2]Cardno, Wellington, New Zealand
*Email: tony.moore@otago.ac.nz

Abstract

The Regular Hierarchical Surface Model (RHSM) is a functional, multi-scale, regular, implicit model developed to support hydrological modelling. The core concept of the RHSM is the ability to model parameter variability with scale explicitly using a hierarchical data structure. The surface model utilises an adapted version of the Hexagonal Image Processing (HIP) addressing system that was generalised for the regular tessellations: triangular, hexagonal and rectangular. A Level of Detail (LOD) model is used to generate variable density realisations by combining pyramid layers generated using a scaling rule with error values and a decision rule. Here we present the definition, implementation and initial assessment of the RHSM.

Keywords: surface modelling, multi-scale, regular tessellation, Level Of Detail (LOD), hydrological modelling.

1. Introduction

There has been a convergence of approaches to modelling hydrology in natural and urban areas, with spatial arrangement, network structure and sub-area behaviours being increasingly taken into account (Fletcher et al, 2013) facilitated by availability of fine data (e.g. LiDaR). This promises to meet a need to provide the complexity required to accurately represent hydrological features at the feature, neighbourhood and city scale (Wright and Leonard, 2012), whilst retaining simplicity of structure and efficiency (Wilson and Gallant, 2000). Recent examples that have sought to explicitly address the issues of scale in urban hydrology include triangular unstructured meshes (Tsubaki and Fujita, 2010) and multi-level coarse grids (Chen et al, 2012).

The *Regular Hierarchical Surface Model (RHSM)* presented here extends and advances these developments by utilising techniques from LOD models. These generate realisations of surfaces using hierarchical data structures, adapting their resolution to the hydrological significance of the underlying area (utilising the geomorphometric parameter flow direction). The RHSM applies LOD modelling within an implicit hierarchical referencing system that is a generalisation of the HIPS 7 hexagonal addressing system to support a choice of one of all three regular tessellations – rectangular (square), triangular and hexagonal (hexagons can be advantageous for hydrological modelling – de Sousa et al, 2006).

2. Definition of the Regular Hierarchical Surface Model

2.1. Indexing

Middleton and Sivaswamy (2001) described the Hexagonal Image Processing (HIP) system which included a linear, hierarchical indexing method called the HIP index. The HIP index is only applicable to aperture-7 hexagonal hierarchies (Wright et al, 2014), whereas the RHSM extends the HIP indexing system to aperture-9 rectangular or aperture-4 triangular hierarchies. Therefore in the context of the RHSM, the abbreviation "HIP" is generalised from Hexagonal Image Processing to *Hierarchical Image Processing (HIP).*

A *HIP* ordinate consists of a sequence of numbers that represents the location of a specific value within a *HIP* dataset. The individual digits are read from left to right from coarsest geographic scale (or tessellated units) to finest. The example in Figure 1 is a 3-level (3-lambda or 3λ) HIP^7 ordinate, 043, where 0 represents the coarsest (largest) hexagon, 4 the intermediate size and 3 the finest.

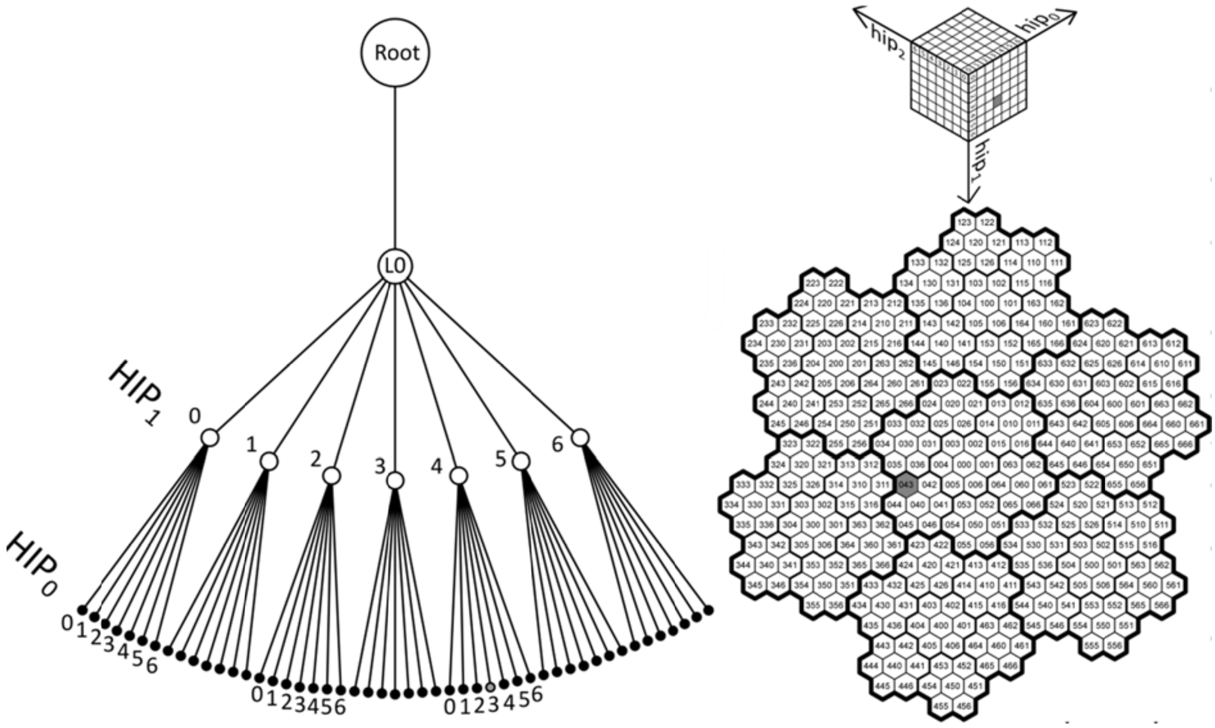

Figure 1: The tree (shown on the left) and array (shown on the right) components of the HIP^7 referencing system. The grey node in the tree and grey square corresponds to the grey hexagon and has HIP^7 reference 043.

2.2. Array tree structure

The *HIP*-based multiscale model uses both trees and arrays (Fig.1). *Tree* structures conveniently store a nested hierarchy where the branches represent the decomposition of space into finer parts via a regular subdivision. This storage method is very similar to a quadtree except that there are seven, nine, or four branches that divide space into hexagons, rectangles or triangles respectively. The tree data structure has regular, sparse or variable density types.

However, larger trees can be inefficient if they have many levels of branching due to the need to store several links between nodes for each datum. This inefficiency can be reduced by tiling, where rather than placing individual values in memory, *arrays* of values that are spatially adjacent can be stored in a tree structure (Platings and Day, 2004). This need not sacrifice the integrity of the hierarchical structure because the array can employ an equivalent hierarchical referencing system to the tree. The RHSM stores values in arrays attached as leaves to a tree data structure. This is similar to the tree and tile division used by Platings and Day (2004) and the Bing Maps Tile System (Microsoft, 2017).

2.3. Level of Detail Modelling

An appropriate scale is selected in three steps using LOD modelling (De Floriani et al. 2005):

- A hierarchical pyramid is formed by a scaling function. If the sampling scale or measurement scale of the parameters in a model differ from the model grid scale, interpolation or generalisation are required, which can be accomplished with a scaling function.
- An error value, the extent to which the generalised value disguises the underlying variation, is associated with each value in the pyramid, as defined by the scaling function.
- A decision rule is then evaluated against the error value to identify the appropriate scale. A simple example is to set a threshold which represents the largest acceptable error value.

An LOD realisation is a variable resolution surface that has been generated from a multi-resolution dataset, typically in real time for visualisation. In the RHSM case, this is adaptive, generated from the coarsest cells that satisfy the decision rule. The adaptive resolution realisation is held in a sparse tree structure. There is no array component of the sparse tree because arrays imply a regular distribution. It is the values in the sparse tree that are then used for subsequent variable density hydrological modelling.

The decision rule can consider not only underlying data values such as elevation but also derived parameters such as flow direction. Thus, LOD modelling is extended here to distributed hydrological modelling.

3. Implementation and Initial Assessment

3.1. RHSM Overview

There are four components of the RHSM computational model, the datafile, the GIS representation of the data, the user and an object class encoded in Python that connects the other three parts. The datafile stores the spatially distributed parameters as NumPy arrays, which can be accessed in a Python script using the PyTables package. The RHSM was integrated with the commercial GIS Esri ArcGIS, where the data (as Feature Classes) was visualised and interacted with by the user (also facilitated by Python).

The RHSM can support very large datasets. The task of managing the nodes in memory is undertaken by PyTables, which retains the most recently used nodes. The array component must always be small enough to be stored in main memory. The tree component of an RHSM dataset is generated using an intermediate array. This places an upper limit on the tree size which is similar to the array size. Although the exact size of the array that can be used will depend on available computer power, Figure 2a indicates that the theoretical maximum is unrealistically large for a commodity computer. Peak performance occurs around an aggregation value of 4 for small RHSM datasets and then increasing with λ after that.

A manageable but still large file size of 100 GB would comfortably contain a λ11*HIP*[7] dataset. However, datasets of this cardinality would be impractically slow to process on a commodity computer. Therefore, processing speed is the practical restriction on RHSM dataset size, not main memory. Figure 2b indicates the extent and level (λ) of RHSM datasets required to cover central Dunedin, New Zealand with 10 cm cell size, which is a resolution common in ortho-imagery and achievable in DEMs generated by LiDAR or drone based photogrammetry.

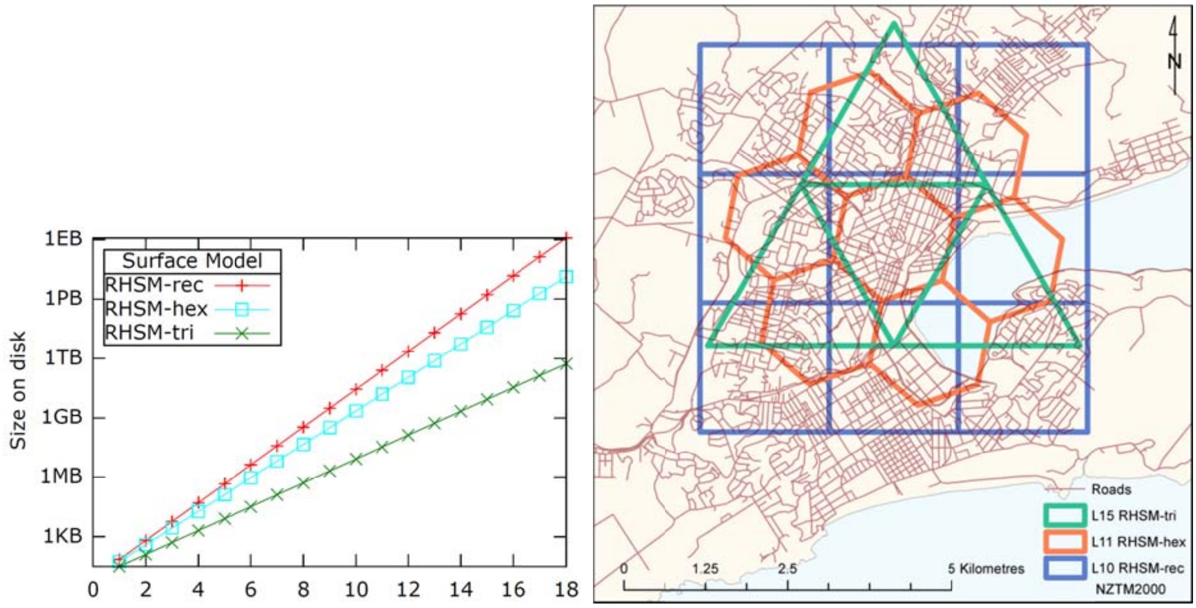

Figure 2: a) Size of RHSM datasets on disk. The size represents the size in memory of a 64bit float NumPy array. (b) Extent of RHSM datasets covered by comparatively large (but still less than 100GB) RHSM datasets. L0 cells have a cell size of 10 cm (not shown). λ-1 shown. The extent of the datasets is compared to the Dunedin urban area.

5. Summary and conclusion

The RHSM is a functional, multi-scale, regular model, with implicit geometry, shape and hierarchy, using an adapted Hierarchical Image Processing (*HIP*) system for indexing. Fundamentally, it supports a choice of three regular geometries: Triangular, RHSM-tri; Rectangular, RHSM-rec; or Hexagonal, RHSM-hex. It employs a regular discretisation of both space and scale, supporting scale variation. This implies that it supports different values for the same area at different resolutions (pyramids) and it generates pyramids with a scaling function. It can also generate level of detail realisations (variable density) using an error value and decision rule. Finally, its data model is an array tree where the array structures are leaves of the tree, where the tree / array threshold is user defined.

Potential refinements to the RHSM conceptual model include:

- applying it to a global referencing frame and relaxing the hierarchy geometry to allow a greater range of resolutions.
- introducing fractional *HIP* addresses to provide finer than base scale resolution in restricted areas and allowing branching at these fractional scale ratios.
- improving the performance speed of the RHSM computational model, for example by utilising more efficient languages, i.e. a C-variant.
- adapting flow accumulation and direction algorithms so that they are generalised for variable resolution surfaces.

6. References

Chen, A S, Evans, B, Djordjevic, S and Savic, D A. 2012. Multilayered coarse grid modelling in 2D urban flood simulations. *Journal of Hydrology*, 470, 1–11.

de Floriani L., Kobbelt L., Puppo E. 2005 A Survey on Data Structures for Level-of-Detail Models. In: Dodgson N.A., Floater M.S., Sabin M.A. (eds) *Advances in Multiresolution for Geometric Modelling. Mathematics and Visualization,* Springer, Berlin, Heidelberg.

de Sousa, L, Nery, F, Sousa, R and Matos, J. Assessing the accuracy of hexagonal versus square tiled grids in preserving DEM surface flow directions. In M. Caetano and M Painho, (eds.), *Proc. Accuracy 2006*, Lisbon, 191-200.

Fletcher, T D, Andrieu, H and Hamel, P. 2013. Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art. *Advances in Water Resources*, 51(0):261–279.

Microsoft. 2017. *Bing maps tile system*. Accessed 30 April, 2017. URL: http://msdn.microsoft.com/en-us/library/bb259689.aspx.

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

Platings, M and Day, A M. 2004. Compression of large-scale terrain data for real-time visualization using a tiled quad tree. *Computer Graphics Forum*, 23(4):741–759.

Tsubaki, R and Fujita, I. 2010. Unstructured grid generation using LiDaR data for urban flood inundation modelling. *Hydrol. Process.,* 24(11):1404–1420.

Wilson, J P and Gallant, J C. 2000. *Terrain analysis: principles and applications.* Wiley, 2000.

Wright, J W and Leonard, G H. 2012. Evaluation of automated flow direction algorithms for defining urban stormwater catchment boundaries. *New Zealand Surveyor*, 301, 28-38.

Wright, J W, Moore, A B and Leonard, G H. 2014. Flow direction algorithms in a hierarchical hexagonal surface model. *Journal of Spatial Science*, 59, 2, 333-346.