Partitioning Techniques to Make Manageable the Generalisation of National Spatial Datasets

Omair.Z. Chaudhry & William.A. Mackaness
Institute of Geography,
School of Geosciences,
The University of Edinburgh
Drummond St
Edinburgh EH8 9XP

Abstract

Map generalisation is a modelling process in which it is typical that detailed, high dimensional geographic phenomena are reduced down to a set of more generalised concepts (for example that a large cluster of buildings is reduced down to the higher order concept of 'city'). This process of generalisation necessarily requires us to handle very large volumes of data which results in high processing overheads. One way of managing high processing overheads is to partition the data. When we break a dataset up into chunks, we need to partition it in such a way that each partition can be generalised without having to consider regions outside that partition. This paper illustrates how the shape and form of partitions are governed by the phenomena being generalised and the type of generalisation methodology being applied. We argue that there is no single partition set suitable for all map features. Instead one can envisage a library of partition sets derived from different data types (combined or applied in isolation) that can be used to support various generalisation processes. The ultimate ambition is to make the problem sufficiently manageable that we can create generalised descriptions at the National Scale (say for Great Britain) from the very detailed. The paper presents various examples of partition sets and illustrates how these can be used to generalise national coverages.

1.0 Introduction

Many National Mapping Agencies maintain detailed, large-scale mapping at national coverage with plans to add increasing amounts of detail and dimensionality. Consequently the data volumes are increasing, as are the sophistication of 'context aware' generalisation algorithms. The processing overheads associated with data enrichment, analysis and generalisation of national coverages require us to give careful thought as to how we can manage and efficiently handle very large volumes of data – particularly if we are to create multi resolution databases capable of automated update. For example it is estimated that the Ordnance Survey's MasterMap® database contains over 400 million objects. Map generalisation is a well understood topic (Buttenfield & McMaster, 1991; Mackaness *et al.*, 2007; Müller *et al.*, 1995). Over the past three decades, a great deal of effort has gone into developing geometric based algorithms that can be applied to specific classes of features (algorithms for point, line and areal feature generalization (Regnauld and McMaster (2007)). Typically such algorithms have high processing times and require large amounts of memory (Boffet & Serra, 2001; Li *et al.*, 2004). Their utility is often demonstrated using small datasets, and the scalability of these methods are rarely considered (Wu *et al.*, 2007). From a pragmatic point of view, we argue that before we can apply these algorithms it is first necessary to partition the dataset. Successful partitioning of the data enables us to take

advantage of both concurrent programming environments (Oaks & Wong, 1997) and GIS systems that support parallel processing.

2.0 Partitioning

There is a considerable literature on the topic of partitioning geographical data (Goodchild, 1989; Han *et al.*, 2001; Harel & Koren, 2001; Samet, 1989; Sloan *et al.*, 1999), most notably methods for partitioning graphs (Tu *et al.*, 2005; Wu & Leahy, 1993), quadtrees, R trees (Kothuri *et al.*, 2002; Mark, 1986) and GAP trees (van Oosterom, 1995; van Putten & van Oosterom, 1998). The dendrogram generated from clustering spatial data is an intuitive way of visualising how we can partition data, and the Voronoi and its dual, have also been proposed as methods for the efficient indexing of spatial information (Yang & Gold 1996; Zhao *et al.*, 1999). Many of these techniques have been used as a basis for creating hierarchical structures – critical to the retrieval and dynamic rendering of geographical information (Frank & Timpf, 1994; van Oosterom & Schenkelaars, 1995) - (as is required in virtual fly throughs for example). Such structures are relevant to the field of generalisation, where we wish to store multiple representations of objects at varying granularity. Partitioning datasets in order to balance loads among multi processors is an important ambition. We would argue that because of the dependency between the 'geography' of an object and generalisation, that it is critical that we partition datasets in a way that takes into account the geography of the object being analysed. Using geometric partitioning that fails to reflect the geography of the features (in particular their interdependence with other features) can result (inadvertently) in artefacts of the partitioning process becoming part of the geography of those features. For example, some algorithms for point thinning of lines (such as Douglas & Peucker (1973)) require a start or anchor node to be specified. If linear features are geometrically partitioned and independently processed, then the point at which those features are 'broken' will affect the resulting output. In the example of Figure 1, the generalised lines (b) vary depending on where the partition falls. In this instance, a far better solution would be to partition the line according to its complexity or characteristic components (Figure 1c). By doing so, a more robust solution is derived. Additionally if we partition according to the character and geography of the line, then we have the opportunity of applying different algorithms according to the nature and character of the feature being generalised (Plazanet, 1995).

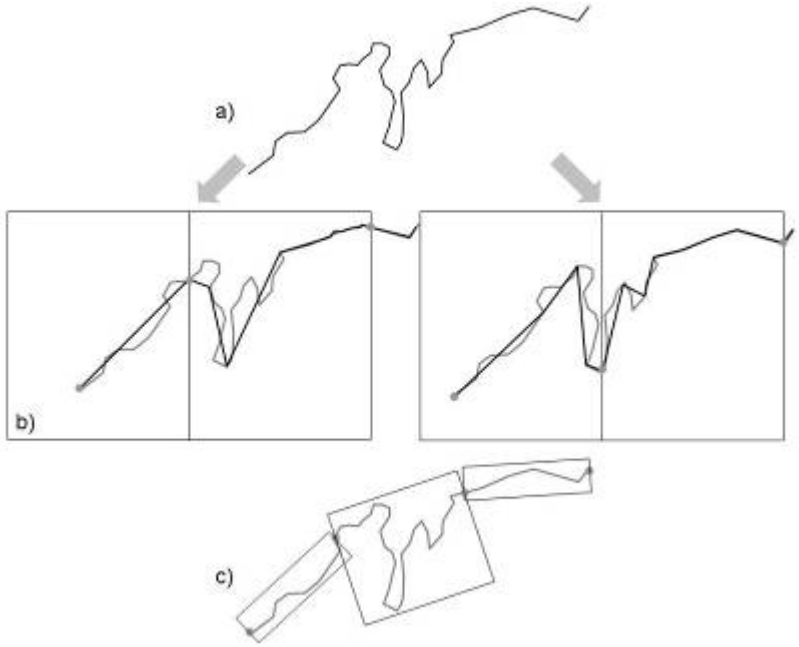

Figure 1: For the input (a), changing the location of the partitioning tile generates different results (even though the parameters of the algorithm remain the same) (b). A more meaningful and useful solution is to partition according to the geography or character of the object (for example c).

Figure 2 illustrates a similar idea but with respect to point objects. In Figure 2a, partitioning is by simple tiling (not taking into account the manner in which the data are clustered) – each tile partition has been processed and the results recombined resulting in the creation of three convex hulls. We observe that the creation of the convex hulls is an artefact of the tiling. In Figure 2b the road network has provided a useful mechanism by which the data can be partitioned (in this case) – resulting in the creation of convex hulls that reflect the distribution of the phenomenon (Figure 2b).

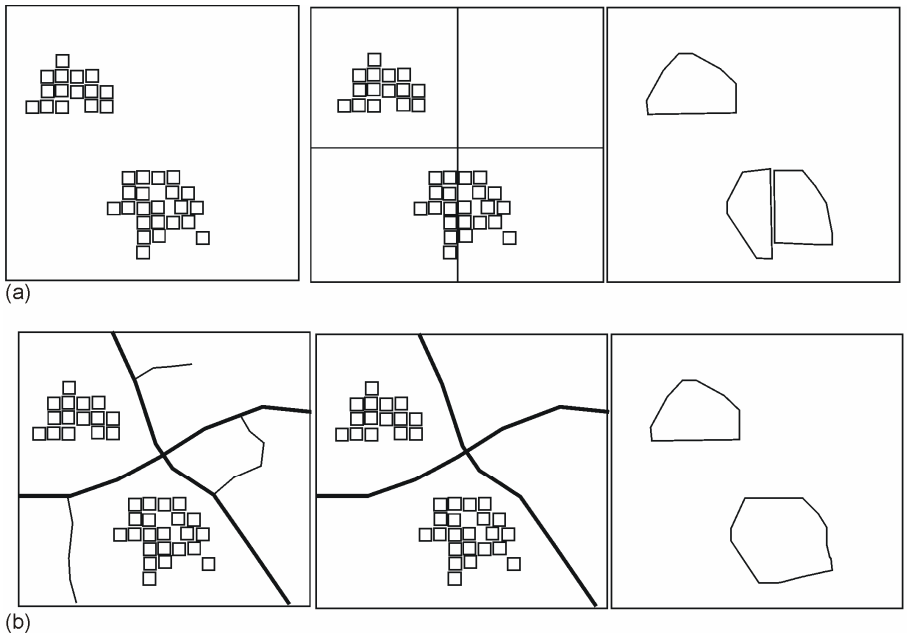

Figure 2: A simple task (creating convex hulls) can be affected by the way the data are partitioned. In 2a partitions are creating using fixed size tiling. Whereas in Figure 2b we have partitions based on part of the road network classification – a situation in which we have been able to ensure that the convex hull reflects the underlying phenomenon being mapped.

2.1 'Meaningful' Partitions

We know that geography is not as well behaved as these two examples might suggest. Thus care must be taken in the creation of these partitions. The partitions must be 'meaningful' in the sense that their creation needs to be sensitive to the geographic nature of the features being generalised, their distribution and their interdependencies. For the purposes of this discussion it is useful to consider the map as being made up of three types of objects: networks, space exhaustive tessellations (surfaces comprising deformable objects) and discrete (small) rigid objects. Each of these data types can be used as a basis for creating different partition sets. Furthermore we observe that typically there are strong correlations between various geographic phenomena – for example there is a strong correlation between the density of road junctions (point objects) and the density of buildings (area objects) or that the morphology of the landscape (a surface) defines catchments in which we will find river networks. Given these sorts of interdependencies, we might use cluster analysis of point data of junctions to define partitions for the generalisation of groups of buildings, and use morphological analysis (Wood, 1996) to create a set of partitions for handling the generalisation of river networks (which themselves would be modelled in graph theoretic form (Gibson *et al.*, 2005; Kawaji *et al.*, 2001) as a basis for 'pruning' the network (Ai & Liu, 2006; Miller & Shaw, 2001; Thomson & Richardson, 1999). Each approach will generate a different set of partitions. These partitions provide a basis for the 'rough' division of the phenomena being generalised. In this sense they are akin to the tailor making a suite, the challenge is in 'roughly' dividing the national coverage into partitions, and then applying a particular process to the objects contained in each partition. Our focus in this paper is on generalisation but different processes may well require different partition types. Figure 3 seeks to summarise these ideas – that various data types (Figure 3a) can be used as a basis for creating different types of partitions (b) which themselves can be permutated with other partition sets to create a range of partition sets in order to 'rough cut' the datasets. For example one might develop an algorithm for generalisation of mountainous villages, in which case the partitions from the city and height partitions can be combined to identify possible candidates.

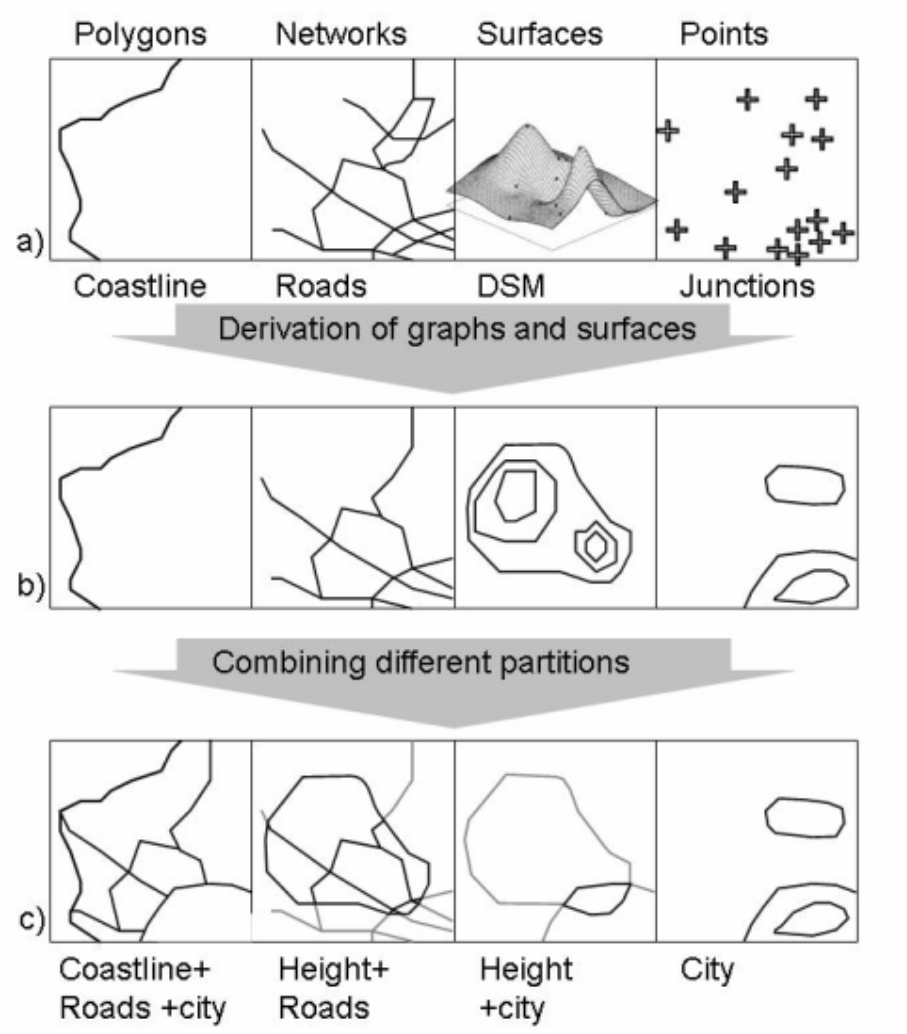

Figure 3:Creating and combining partitions in different ways (c) – based on (a) polygons, networks, surfaces and points.

In the following three sections we illustrate the creation and application of different partition sets and demonstrate how they can be applied to national datasets. We demonstrate how this approach makes manageable the handling of very large datasets offering the possibility for concurrent programming. The three approaches demonstrate the importance of partitioning detailed datasets as a pre-process to creation of feature boundaries (prominent hills/ranges, settlement and forest) that are present at lower levels of detail (say 1:250k). The parameter settings used are in anticipation of this resultant level of detail. For other levels of detail these parameters would require tuning.

3.0 Partitioning Digital Terrain Model prior to extraction of Hills and Range boundaries

An earlier paper (Chaudhry & Mackaness, (in press)) presents an approach for the extraction of summit boundaries from a high resolution digital terrain model (DTM). In summary the technique measures prominence (relative difference in height) and morphological variability in the landscape as a basis for defining the extent of a range or a hill. Additionally the algorithm is able to model the 'child parent' relationship between the range and its various subcomponents.

The technique is computationally intensive. If we are to apply this approach to a national coverage, we must first partition the dataset (to manage processing and memory constraints) – but in a way that does not destroy the continuity of any given range. A partition would not be meaningful if it split a hill or range in half – since in separately processing the two halves, we would lose the identity of the hill or range as a whole. This is precisely what would happen if we use regular tiling. What is required is a partitioning that divides the country up into broad morphological regions, and then for the generalisation algorithm (Chaudhry & Mackaness, (in press)) to then be applied within each of these broad regions.

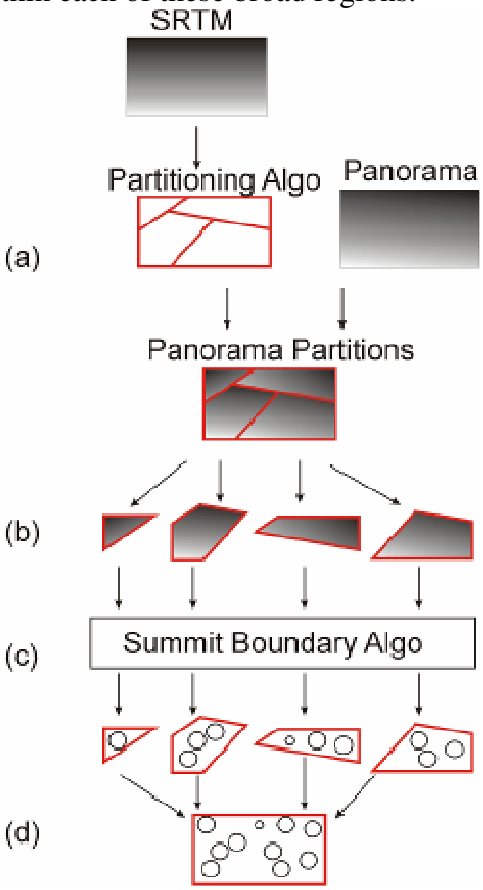

Figure 4: We use a low resolution DTM (SRTM) to partition a detailed DTM (Panorama) and create summit extents in each partition before re-assembly of the partitions.

The overall methodology for partitioning and generating of hills and range boundaries from DTM can be viewed as a combination of three sub methodologies (Figure 4). The first stage of the methodology uses a low resolution DTM (SRTM data 90m) for the creation of the partition set (Figure 4a). These partitions are used to 'cookie cut' regions from a detailed DTM (Ordnance Survey Land-Form Panorama® data 50m or Land-Form Profile Plus® 10m). A summit boundary detection algorithm (Figure 4c) is then applied to each partitioned DTM region (Figure 4b), before being 're-assembled' back into a single file (Figure 4d).

3.1 Partitioning Methodology Using Surfaces

We begin by describing the partitioning methodology (Figure 4a). In a generalisation context, computational effort is greatest in regions where there is high variance among complex morphology (in relatively flat regions the computational effort is much less). The methodology of partitioning separates regions of high variability from low variability regions. We use as input, low resolution data from the Shuttle Radar Topography Mission (SRTM) – a

mission that collected a digital terrain model for 80% of the world (www2.jpl.nasa.gov/srtm/). The SRTM covering most of Scotland is shown in Figure 5. The partitioning algorithm begins by creating a relief surface. Relief is the difference of elevation of each location as compared with its surrounding within a prescribed locus (Summerfield, 1991). The relief for each location (pixel or cell) of the SRTM data is calculated by searching for the highest and lowest point of the relief within a passing circular kernel of given size. The larger the size of the kernel the more neighbouring cells considered for each cell. In this research the radius was empirically determined and was set to 10 cells. Each kernel thus included a region of 900m by 900m. More than 10 cells and the relief is of a very general form; less than 10 cells and the relief is unnecessarily detailed for the purposes of creating our 'rough cuts'. The resulting relief surface for the SRTM DTM in Figure 5 is shown in Figure 6. We then created a masked surface using a relief threshold (empirically determined to be 60m). This value was chosen because we are only interested in significant changes in the landscape. If the relief value for the cell is above or equal to this threshold then it is assigned a value 1, otherwise 0. The resulting surface is then converted into polygons (using the raster to polygon utility in ArcGIS 9.2). All adjacent cells that have the same value (0 or 1) are grouped into the same polygon. Figure 7 shows the three resulting partition polygons for Scotland. Some of the resulting partition polygon are quite small in extent and represent a small region they can either be processed separately or can be processed as part of the residual (grey polygon in Figure 7). Figure 8 shows the application of same process with same threshold values using SRTM for all of the UK and Ireland.

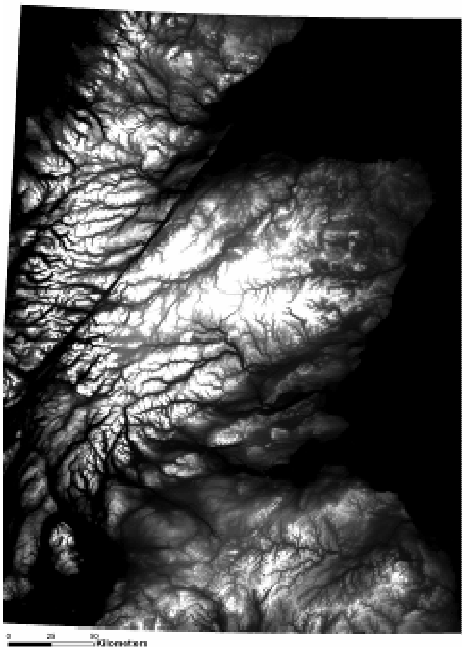

Figure 5: SRTM data for Scotland

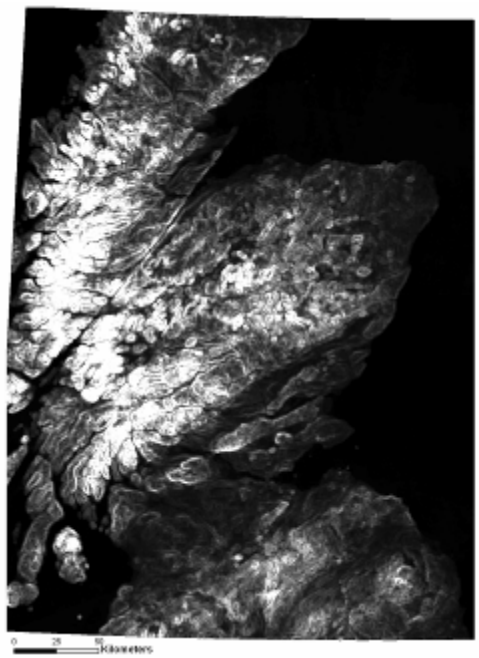

Figure 6: Relief Map of Scotland using SRTM

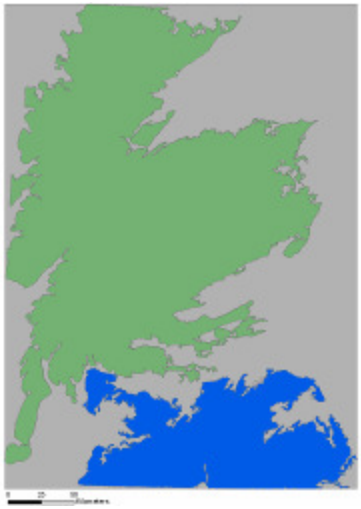

Figure 7: Partitions created using the partitioning algorithm and SRTM data

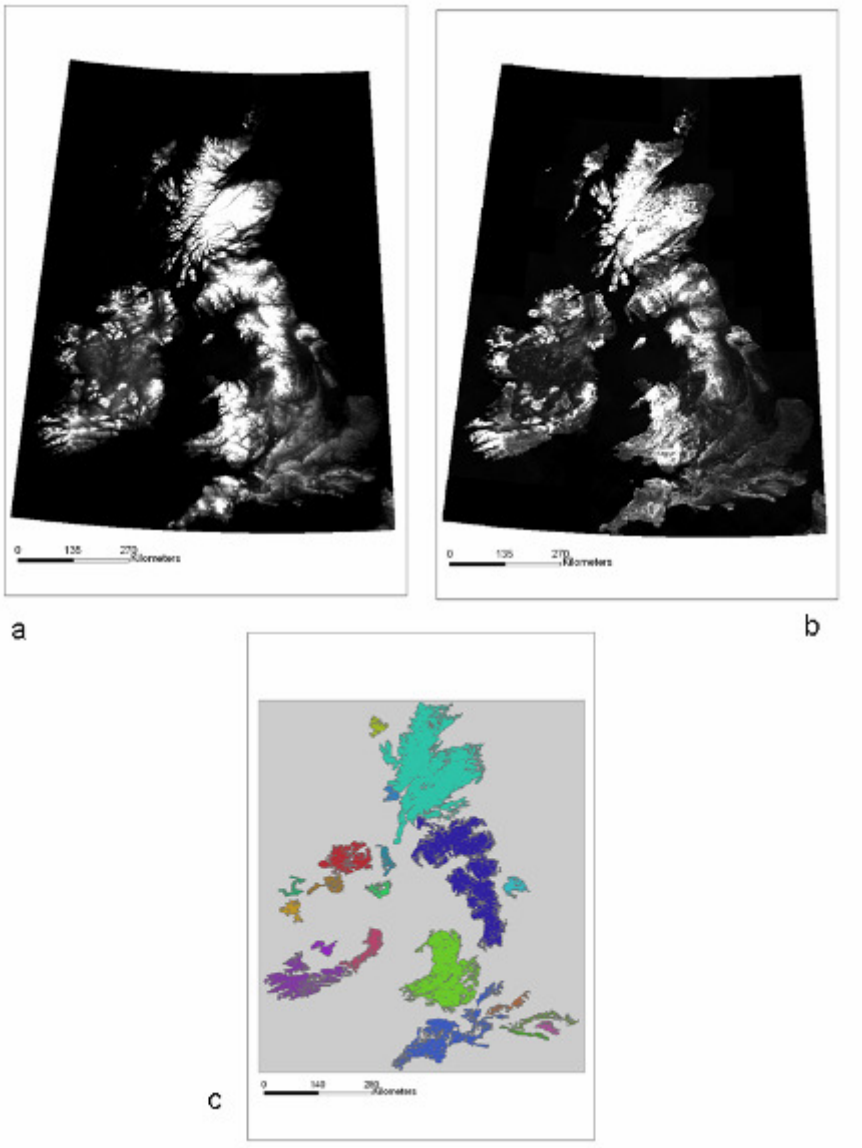

Figure 8: a) SRTM for UK and Ireland b) Relief from SRTM c) Partition polygons (21) for theUK and Ireland

The partition polygons are then used to partition a high resolution DTM (OS Land-Form Panorama), a 50m DTM that covers the whole of Great Britain. Thus the DTM in Figure 9a was intersected by the partition polygons of Figure 7 resulting in the division of the high resolution DTM into three sub DTMs (Figure 9b). These three DTMs were processed separately in order to identify the hills and range boundaries (Figure 4c). The resultant boundaries obtained within each partition by this process (Chaudhry & Mackaness, (in press)) are shown in Figure 10.

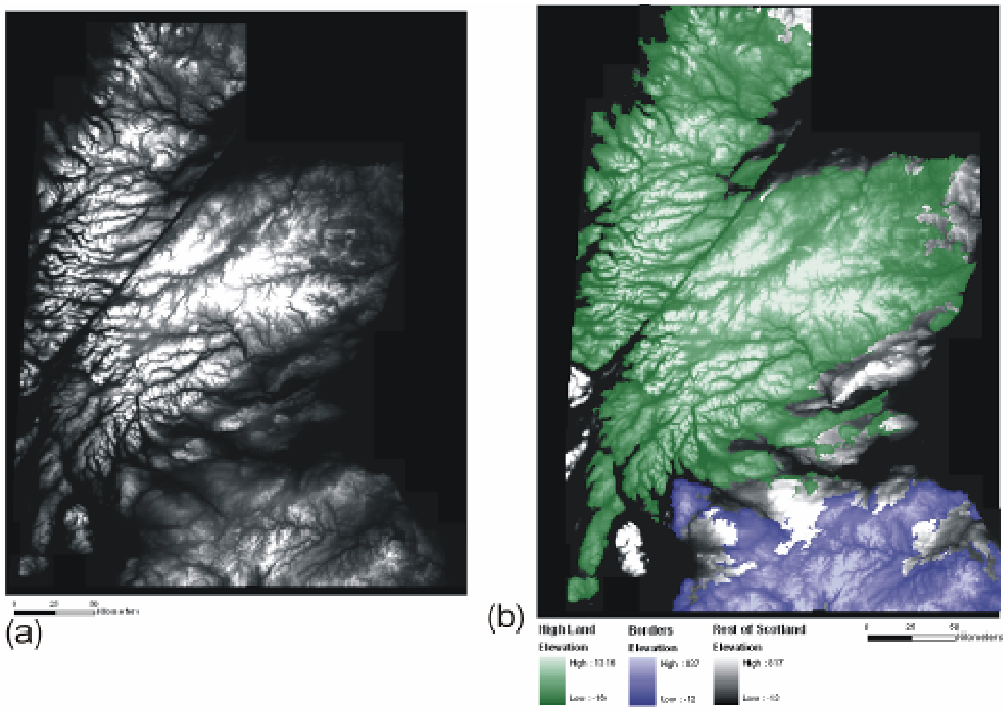

Figure 9: (a) OS Land-Form Panorama (50m) DTM (b) Source DTM (Figure 9a) partitioned into 3 sub-DTMs (Ordnance Survey © Crown Copyright. All rights reserved)

Figure 10: The reassembled Hills and Range boundaries using the three DTM partitions shown in Figure 9b

Table1 shows the processing times for summit boundary algorithm and for the creation of partitions from SRTM. It also shows the total number of summits found in each partition and the time taken to process each partition. The whole process (Figure 4) was executed on a standard PC (2.99GHz, 2 GB of RAM). As an additional part of the evaluation, the correctness of the boundaries shown in Figure 10 were evaluated by calculating the distance of each hill and range boundary from the text points taken from the 'Land Use' layer of Ordnance Survey's Strategi® dataset (at 1:250K). The statistical analysis of text points from OS Strategi and their distances from the boundaries showed that nearly 95% of all text points are within the resultant extents. A few texts are within a few meters of the extent, but fell outside the boundary. This is mainly due to cartographic operations, such as displacement, applied to the text points in OS Strategi dataset. It is important to note that in Strategi there is no link between the text points and the objects they represent but with this approach, the boundaries can be associated with the text, thereby used to improve automated text placement. Additionally these extents can form the basis for other types of spatial analysis and enrichment of other topographic databases (Chaudhry, 2007).

Table 1: Processing time for hill and boundary dectection algorithim for each of three partitions. Also the processing time for creation of partitions using SRTM.

Data Set	Region	Number of Summits	Total Time	Time per Summit in mins
Creation of Partitions using SRTM			20 mins	
Land-Form Panorama (50m)	Borders	1,613	95 mins	0.06
Land-Form Panorama (50m)	High Lands	6,046	29 hrs	0.28
Land-Form Panorama (50m)	Rest of Scotland	438	27 mins	0.06

4.0 Partitioning Building Data Prior to Settlement Boundaries

By way of a second example, we illustrate how partitions can be formed based on the density of point data. In this example, the focus is on creating a set of partitions for handling a very large numbers of discrete areal features. The focus is on generalisation of buildings in order to create the boundary to a city or settlement represented at 1:250K. If we consider only Scotland, this requires us to handle approximately three million buildings as polygons. The point data used to create the partition set is the nodes from the road network – stored as part of Ordnance Survey's ITN dataset. The junctions are single node points, whereas the buildings are described as complex polygons. Using point data is far more computationally efficient. Using this approach we can derive products directly from the very fine detailed 1:1250,1:2500, 1:10K database and produce generalised results for representation at notional scales of 1:250K.

The settlement boundary generation methodology has been explained in an earlier paper (Chaudhry & Mackaness, 2008). In brief, it involves calculating a 'citiness' value for each building based on its size and density, creating clusters based on density of buildings, amalgamating them in order to define a boundary and selected of boundary significant for 1:250K level of detail. Our focus here is to explain the partitioning methodology used prior to the execution of the settlement boundary algorithm.

4.1 Partitioning Methodology Using Point Data

Such large volumes of data not only require large storage space but also require large amounts of memory and processing time in order to identify settlement boundaries. The requirement is to partition the data such that building data can be handled separately within each partition without affecting the resultant settlement boundaries. Various ideas were explored but one candidate that proved very useful in the creation of partitions was to use road nodes selected from OS MasterMap's Integrated Transport Network (ITN) dataset. The reason being that there is a strong association between the road network and building objects. Where there is a high density of buildings (a settlement or town), there is high density of roads (and vice versa). These road objects in ITN are topologically structured in terms of graph theoretic elements i.e. segments and nodes (Beard & Mackaness, 1993; Molenaar, 1998) . Each road segment has a start node and an end node. These were input into a clustering algorithm in order to create partition boundaries. Road nodes are represented as a single point (dimension 0), rather than buildings (which have complex area geometries); there are far fewer road nodes than buildings (of the order of 1 node for every 10 buildings), and the road nodes are closely correlated with buildings both in urban and rural areas (in effect where you find buildings you find road nodes and vice versa). This is summarised in Table 2.

Table 2: The number of buildings per node for an extended region around Glasgow

	Buildings	Nodes	Buildings per node
Urban regions	310,502	30,267	10.3
Rural areas	30,900	3,188	9.7
Overall	341,402	33,455	10.2

Road nodes were selected for the entire region of interest (in this case for the whole of Scotland; total road nodes 394,696) and were loaded into a spatial database. For each junction we counted and recorded the number of nodes within a radius of 'x' meters. Different distance threshold were tested ranging from 100m to 3000m. Small thresholds resulted in partitions that were too small for the required level of detail (the same settlement boundary fell within separate partitions). Threshold that were large resulted in large partitions which would result in selection of more buildings thus increasing the processing time. 1000m was found to be the most appropriate because it created large partitions within which groups of buildings naturally fell (without the partition acting to divide up cities or towns). This gave us a measure of density, assigned to each node. We then created a buffer around each node, the size of the buffer being proportional to its density. The overlapping regions were then merged. Figure 11 summarises this process.

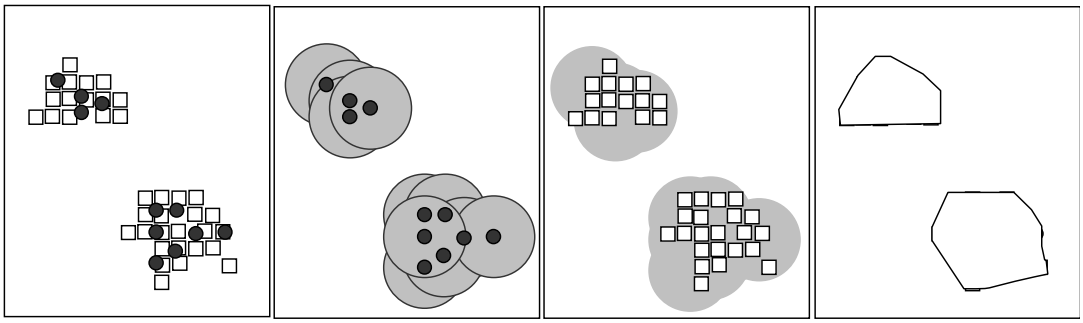

Figure 11: Squares represent buildings; black dots – road nodes. Junction points are buffered and merged to create partitions (grey regions); these are subsequently used to partition sets of buildings which are then processed (in this figure, to produce simple convex hulls).

Figure 12 shows the partition set created using this approach for the whole of Scotland. The right hand side of Figure 12 shows one partition in more detail. Note that the buildings fall within the partition. The creation of these partitions now affords a means of using concurrent (parallel processing) or sequential processing to analyse and process the whole dataset. The settlement boundary algorithm (Chaudhry & Mackaness, 2008) can now be applied, partition by partition. Figure 13a shows the result of applying the settlement boundary algorithm to the buildings contained within the partition shown in Figure 12. By way of comparison, Figure 13b shows overlayed, the equivalent cartographically hand drawn result for the same region. More on evaluation of resultant settlement boundaries can be found in Chaudhry and Mackaness (2008).

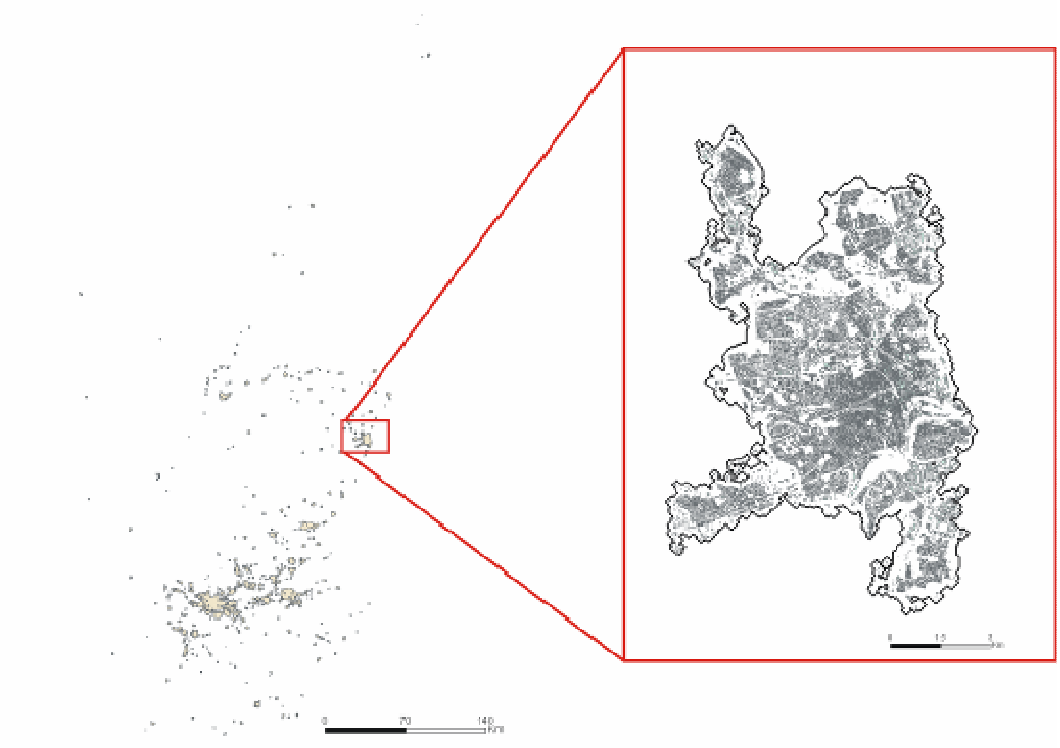

Figure 12: Partitions for entire Scotland created using node dataset with distance threshold of 1000m. The partition is around the city of Aberdeen (Ordnance Survey © Crown Copyright. All rights reserved)

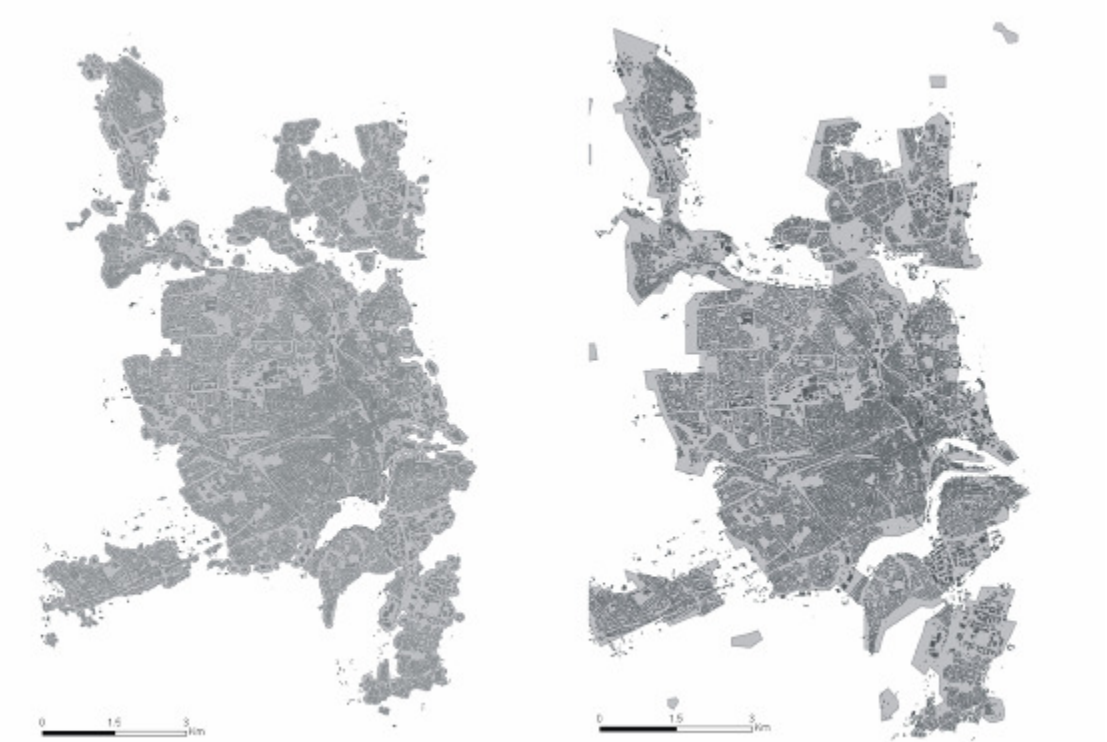

Figure 13: (a) Resultant settlement boundaries derived from partition shown in Figure 12; (b) OS Strategi settlement boundaries (1:250,000) for the corresponding region generated manually by cartographers (Ordnance Survey © Crown Copyright. All rights reserved)

Table 3 summarises the processing times for calculation of partitions and the processing times for calculation of settlement boundaries using the settlement boundary approach (Chaudhry & Mackaness, 2008) for different partitions.

Table 3: Processing times for different stages and partitions

Region		**Total time in Hours**
Partitioning Boundaries (using road nodes for entire Scotland)		1.9
	No of Building	
Settlement Boundaries (Aberdeen region partition Figure12 a)	96,000	2.5
Settlement Boundaries (Edinburgh region partition)	168,830	6.6
Settlement Boundaries (Glasgow region partition)	544,000	22.5
Settlement Boundaries (all remaining partitions)		15.4
The whole of Scotland	3,040,654	(estimate) 98

5.0 Partitioning Tree Data Prior to Forest Boundaries

Having demonstrated how partitions can be created using morphology and point clustering, this third example shows how networks (in combination with polygonal data) can be used to create other types of partition set. In this third example, our focus was on handling the generalisation of large numbers of forest patches. The problem with partitioning forested areas is that there is huge variability in the extent of a forested area, and there is no strong correlation with other feature classes. This is illustrated in Figure 14 which shows tree patches selected from the source database. These tree patches are captured at a high level of detail (1:1250 scale in urban areas, 1:2500 scale in rural areas and 1:10,000 scale in mountain and moorland areas) and are stored as polygons. The objective here is to generate forest boundaries from the combination of these individual tree patches for representation at 1:250,000 scale using a forest boundary detection approach (Mackaness *et al.*, 2008). Prior to construction of these forest boundaries from tree patches, a partitioning approach is required in order to make the problem scalable. Here we present one such partitioning approach.

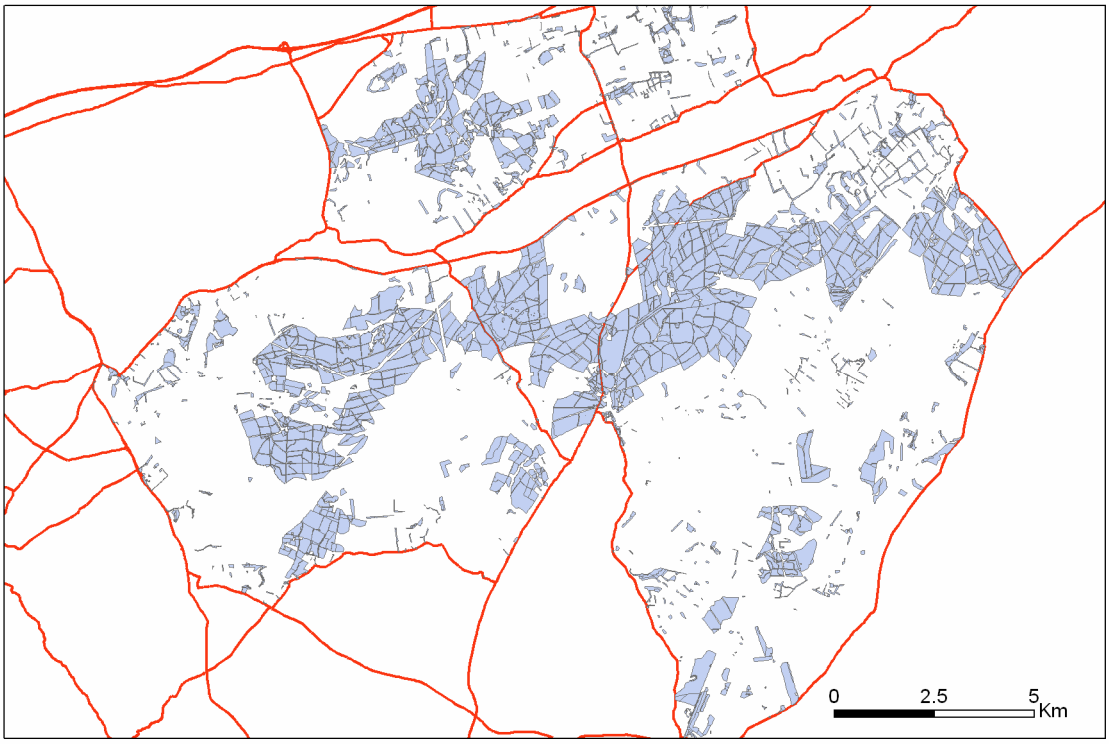

Figure 14: An example of tree patches selected from the source database along with road objects for a region south west of Edinburgh, Scotland. The tree patches overlap with road objects. (Ordnance Survey © Crown Copyright. All rights reserved)

5.1 Methodology

The challenge is in finding a class of feature that can be used as a basis for partitioning tree patches. River networks could be used, but because they are acyclic do not naturally lend themselves to the creation of partitions. It was observed that there is a weak correlation between the location of forestry data (patches) and road partitions in so much as the road partitions form cycles in the graph (closed regions) and 'divide' forest regions. We also observe that the area within a road partition tends to be very small in cities, and cities typically contain little forest. Conversely there are remote parts of Scotland that are poorly serviced by roads (large partitions) yet which are heavily forested. In this project it was decided to examine the role of the road network in partitioning forestry data covering the whole of Scotland. It was not necessary to use all the road classes since this would produce many small partitions, many of which would contain no forested areas. Initial experiments focused on using only 'motorways', 'A' and 'B' roads and subsequent analysis of processing times indicated that this was a pragmatic solution. There was a concern that the road network would be very dense within cities and create very large numbers of small partitions. Given this concern we had considered using city boundary partitions to handle this problem. Though this was indeed the case, the algorithm was very efficient at processing empty partitions and so it was not necessary to used this combining of partition sets. It is also the case that although cities do indeed have high densities of roads, the city itself does not contain especially high numbers of motorways, A and B roads. Therefore cities did not excessively create very large numbers of partitions in the way that we feared it might. Figure 15 is an example of the selection of motorways, A and B roads for a small area and the partitions formed using these selected roads.

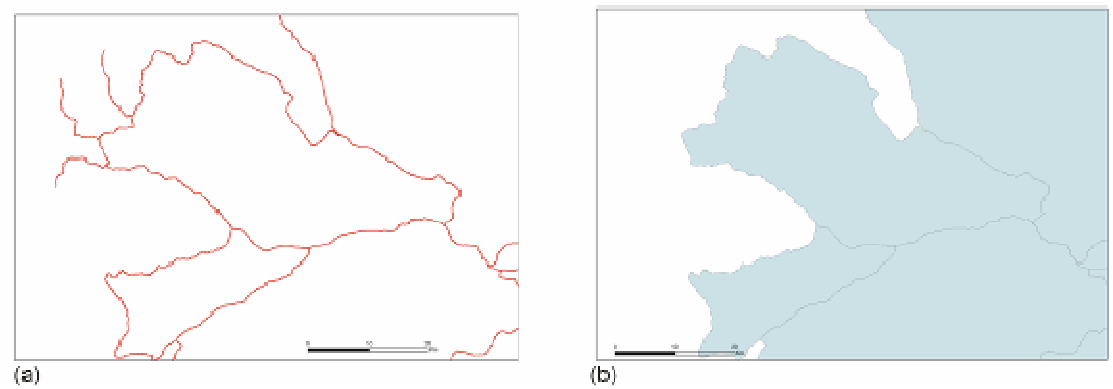

Figure 15: Partitions (b) formed from the selection of motorways, A and B roads only (a). Roads, not reaching the coast, resulting in very large 'open' partitions. (Ordnance Survey © Crown Copyright. All rights reserved)

The creation of a partition requires the graph to be 'closed'. There are many instances of roads that are not closed – like spider's legs they radiate out towards the coast, but do not form closed loops, and do not therefore form a partition (Figure 15). In order to create a set of partitions that covered the whole of Scotland, it was necessary to use the road network, and additionally the mean high water line (MHW) data as a way of 'closing' the partitions at the coastal margins.

Using the MHW was in preference to the coastal line which was broken wherever there was an estuarine feature, and in any case often did not connect with a road – for example where the road stopped short of the shoreline. The MHW is an Ordnance Survey (OS) boundary product, which is also part of Land-Form Profile. The creation of partitions based on the combination of roads and MHW was previously undertaken by the OS as part of an earlier project. Figure 16 shows an example of partitions near the coast and demonstrates how their combined use results in closure of partitions and the creation of an exhaustive tessellation of the land covering Scotland. The selected roads and mean high water features were combined into a single shape file and the partitions were created using a ArcGIS 9.2 Feature to Polygon utility. It took approximately 12 minutes to create the partitions for the whole of Scotland (Figure 17).

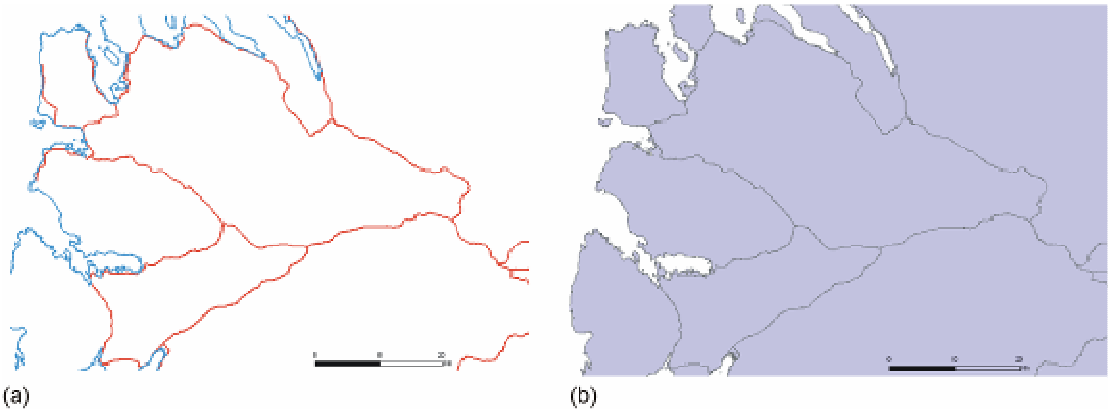

Figure 16: Partitions (b) created using Motorways, A road, B road and MHW data (a). Note that the problem of open partitions illustrated in Figure 15b is removed in Figure 15b by using MHW combined with road data. (Ordnance Survey © Crown Copyright. All rights reserved)

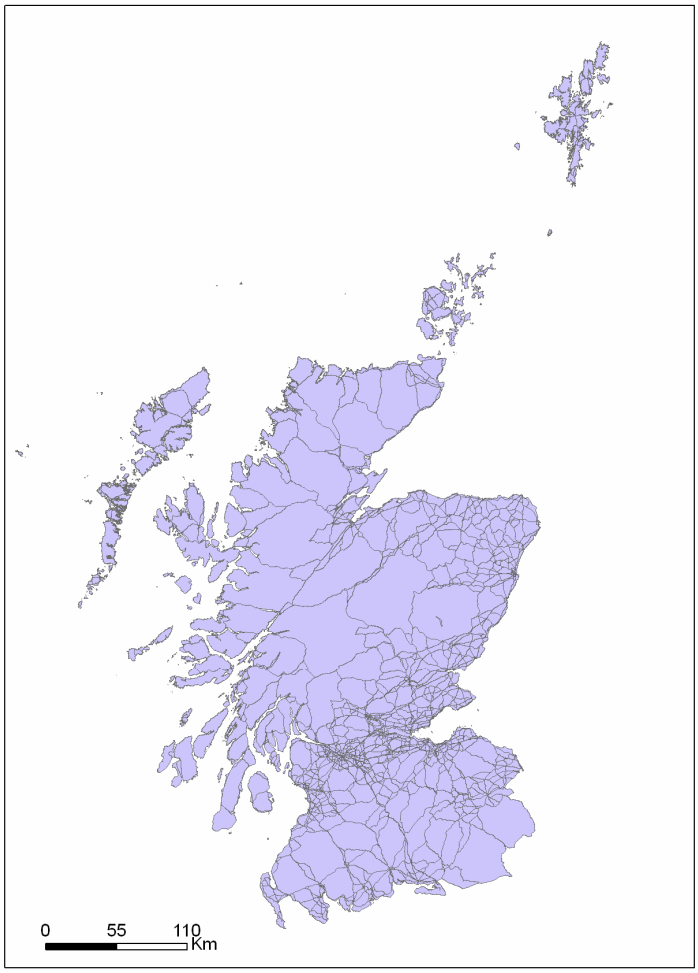

Figure 17: 14 000 Partitions for Scotland derived from important roads and MHW data. (Ordnance Survey © Crown Copyright. All rights reserved)

Combining MHW and roads generated 14,000 partitions for Scotland (Figure 17); the area of each partition ranged in area from 2.67 square meters (a traffic island between a section of dual carriageway) up to a region covering 2,790 square kilometres – a vast remote region south of Fort William. 85% of the polygons were of a size less than 0.1 square kilometres, 13% were of a size between 0.1 and 50 square kilometres, and the remaining 2% were between 50 and 3000 square kilometres.

For the purposes of demonstrating the use of multiple processors, the partition dataset was broken into three partitioning datasets – each sent to a different processor. Partition by partition, the tree patches from the source database for that partition were selected, and processed. It could have been any process, but in this instance the interest was in aggregating tress patches into forest boundaries (Mackaness *et al.*, in press) as illustrated in Figure 18. Figure 19 shows the result of this aggregation process for tree patches shown in Figure 14.

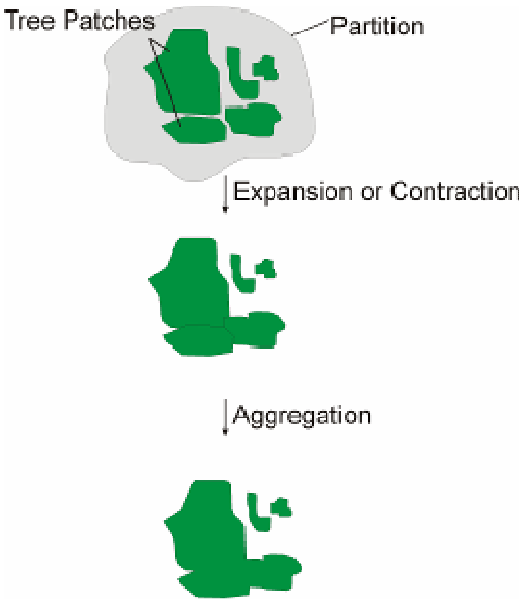

Figure 18: Merging regions together based on area and proximity among a group of tree patches.

Figure 19: Result of aggregation of tree patches for region shown in Figure 14

The time taken to process any given partition was dependent upon 1) the number of tree patches inside the partition, 2) their areal extent, and 3) the complexity of the boundary (the number of vertices used to store the boundary). Many partitions contained no forest, whilst one partition contained 12,127 tree patches. The partition of 12,127 tree patches took just over 60 minutes to process.

86% of the partitions contained no tree patches at all. 9% of the partitions contained 100 tree patches or less, and the remaining 5% of partitions contained between 100 and 12100 tree patches. Thus only 14% of the partitions contained any tree patches. Whilst using these partitions enabled the data to be processed, it indicates that it is far from ideal as a mechanism for partitioning forest data. Though we did not analyse the data, common sense suggests that there is probably a strong correlation between small partitions and the absence of tree patches.

Sometimes a forest region effectively extends across a road (Figure 14 and Figure 19). In this instance the road effectively intersects and divides a forest region. Because of the partitioning process, the objects will be processed independently of each other, if they lie in different partitions. There is therefore a need to ascertain whether forest patches lie either side of a boundary. Thus once all forest boundaries were generated within each partition, it was necessary to check whether any given forest abutting the partition boundary contained a 'neighbour' in the adjoining partition(s). The nature of the aggregation algorithm made this process straightforward in that the aggregation algorithm marginally enlarged each forest patch such that forest patches laying either side of a partition already overlapped (Figure 19). The ArcGIS 9.2 'Aggregate Polygons' utility was used for aggregation of neighbouring (overlapping) forest boundaries, removal of small resultant boundaries and dissolving of small holes. Figure 20a shows the result of this operation. As illustrated the overlapping forest boundaries lying in different partitions in Figure 14 and Figure 19 have been combined into single forest boundaries (as was done by cartographers when creating forest boundaries at 1:250,000 dataset (OS Strategi®) as shown in Figure 20b.

Figure 20: (a) Resulting forest boundaries for tree patches shown in Figure 14. (b)OS Strategi forest data (Ordnance Survey © Crown Copyright. All rights reserved)

Table 4 summarises the processing times for the creation of partitions and resultant forest boundaries for the whole of Scotland. The whole process has reduced the number of tree patches in the source database from 568,662 to 3,755 forest boundaries. This compares with 3347 forest boundaries found in OS Strategi (1:250K). There are two main reasons for this difference. OS Strategi is a cartographic product thus many boundaries that are close have been merged and simplified; secondly OS Strategi is not as current and was created independently from a source database.

Table 4: Processing times

Process	**Time**
Creation of partitions using important roads and MHW for entire Scotland	12 (mins)
Tree patches (total 568,662) into forest boundaries (total 100,443) within each partition for Scotland	29.5 (in hours)
Aggregation and simplification of forest boundaries across partitions from 100 443 to 3 755 for Scotland	13 (hours)

Conclusion

We have demonstrated the use of different partitions as a basis of managing very large datasets. The research indicates that there is not a single ideal set of partitions. The type of partition required will depend on the type of analysis or generalisation intended – each set of partitions variously suitable for partitioning morphology, anthropogenic and natural regions. The research also indicates that combinations of different sets of partitions can increase the efficiency of processing and be used to form stronger correlations between the partitions and the class of feature being processed (more 'meaningful' partitions). 'Meaningful' partitions – ones that account for the geography of the phenomenon can make far more efficient the process of analysis and visualisation.

We argue that by combining partitions we can 1) enrich the database, 2) make greater efficiencies in the handling of data, 3) support the creation of hierarchies, 4) offer innovative ways of visualising GI, and 5) enable more intuitive forms of analysis (linked to the granularity/hierarchy of the data).

References

Ai, T. & Liu, Y. (2006). The hierarchical watershed partitioning and data simplification of river networks. In *Progress in Spatial Data Handling, 12th International Symposium*, pp. 617-632. Springer, Berlin Heidelberg.

Beard, M.K. & Mackaness, W.A. (1993) Graph Theory and Network Generalization in map Design. In 16 ICA Conference, Vol. 1, pp. 352 - 362, Cologne Germany.

Boffet, A. & Serra, S.R. (2001) Identification of spatial structures within urban block for town classification. In 20th International Cartographic Conference, Vol. 3, pp. 1974-1983, Beijing, China.

Buttenfield, B.P. & McMaster, R.B. (1991) *Map Generalization: Making Rules for Knowledge Representation* Longman, London.

Chaudhry, O.Z. (2007) Modelling Geographic Phenomena at Multiple Levels of Detail: A Model Generalisation Approach based on Aggregation. Doctoral Dissertation, University of Edinburgh, Edinburgh.

Chaudhry, O.Z. & Mackaness, W.A. (2008) Automatic Identification of Urban Settlement Boundaries for Multiple Representation Databases *Computer Environment and Urban Systems*, 32(2), 95-109

Chaudhry, O.Z. & Mackaness, W.A. ((in press)) Creating Mountains out of Mole Hills: Automatic Identification of Hills and Ranges Using Morphometric Analysis *Transactions in GIS*.

Douglas, D. & Peucker, T. (1973) Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. *The Canadian Cartographer*, 10(2), 112-122.

Frank, A. & Timpf, S. (1994) Multiple representations for cartographic objects in a multi-scale tree: an intelligent graphical zoom. *Computers and Graphics Special Issue: Modelling and Visualization of Spatial Data in Geographic Information Systems*, 18(6), 823-829.

Gibson, D., Kumar, R., & Tomkins, A. (2005) Discovering large dense subgraphs in massive graphs. In Proceedings of the 31st international conference on Very large data bases, pp. 721 - 732, Trondheim, Norway.

Goodchild, M.F. (1989) Tiling large geographical databases. In Symposium on the Design and Implementation of Large Spatial Databases, pp. 137–146. Springer, Berlin, Santa Barbara, California.

Han, J., Kamber, M., & Tung, A.K.H. (2001). Spatial Clustering Methods in Data Mining: A Survey. In *Geographic Data Mining and Knowledge Discovery*, pp. 1-29. Taylor and Francis.

Harel, D. & Koren, Y. (2001) Clustering spatial data using random walks. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 281 - 286, San Francisco, California.

Kawaji, H., Yamaguchi, Y., Matsuda, H., & Hashimoto, A. (2001) A Graph-Based Clustering Method for a Large Set of Sequences Using a Graph Partitioning Algorithm. *Genome Informatics*, 1293-102.

Kothuri, R.K., Ravada, S., & Abugov, D. (2002) Quadtree and R-tree Indexes in Oracle Spatial: A Comparison using GIS Data In ACM SIGMOD Madison, Wisconsin, USA.

Li, Z., Yan, H., Ai, T., & Chen, J. (2004) Automated building generalization based on urban morphology and Gestalt theory. *International Journal of Geographical Information Science*, 18(5), 513-534.

Mackaness, W., Perikleous, S., & Chaudhry, O. (2008) Representing Forested Regions at Small Scales: Automatic Derivation from the Very Large Scale. *The Cartographic Journal*, 45(1), pp.6-17

Mackaness, W.A., Ruas, A., & Sarjakoski, L.T. (2007) *Generalisation of Geographic Information: Cartographic Modelling and Applications* Elsevier, Oxford.

Mark, D.M. (1986) The Use of quadtrees in geographic information systems and spatial data handling. In Procs.Auto Carto London, , Vol. .1, pp. 517-526.

Miller, H.J. & Shaw, S.L. (2001) *Geographic Information Systems for Transportation: Principles and Applications* Oxford University Press, Oxford.

Molenaar, M. (1998) *An Introduction to the Theory of Spatial Object Modelling for GIS* Taylor & Francis, London.

Müller, J.C., Lagrange, J.P., & Weibel, R. (1995). GIS and Generalization: Methodology and Practice. In *GISDATA 1* (eds I. Masser & F. Salge). Taylor & Francis, London.

Oaks, S. & Wong, H. (1997) *Java Threads* O'Reilly & Associates, Inc, Sebastopol, CA.

Plazanet, C. (1995) Measurement, Characterization and Classification for Automated Line Feature Generalization. In Auto Carto 12 (ed ACSM-ASPRS), Vol. 4, pp. 59-68, Bethesda.

Regnauld, N. & McMaster, R.B. (2007). A Synoptic View of Generalisation Operators. In *Generalisation of Geographic Information: Cartographic Modelling and Applications* (eds W.A. Mackaness, A. Ruas & L.T. Sarjakoski), pp. 37-66. Elsevier, Oxford.

Samet, H. (1989) *The design and analysis of spatial data structures* Addison Wesley, Reading,Massachusetts.

Sloan, T.M., Mineter, M.J., Dowers, S., Mulholland, C., Darling, G., & Gittings, B.M. (1999). Partitioning of Vector-Topological Data for Parallel GIS Operations: Assessment and Performance Analysis. In *Euro-Par'99 Parallel Processing*, Vol. 1685/1999, pp. 691 - 694 Springer Berlin / Heidelberg.

Summerfield, A.M. (1991) *Global Geomorphology* Longman, London.

Thomson, R.C. & Richardson, D.E. (1999) The 'Good Continuation' Principle of Perceptual Organization Applied to the Generalization of Road Networks. In Proceedings of the ICA 19th International Cartographic Conference, pp. 1215–1223, Ottawa.

Tu, J., Chen, C., Huang, H., & Wu, X. (2005) A visual multi-scale spatial clustering method based on graph-partition. In Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International, Vol. 2, pp. 745-748.

van Oosterom, P. (1995). The GAP-tree, an approach to `on-the-fly' map generalization of an area partitioning. In *GIS and Generalization: Methodology and Practice* (eds J.C. Müller, L.J. P & R. Weibel), pp. 120-132. Taylor & Francis, London.

van Oosterom, P. & Schenkelaars, V. (1995) The development of an interactive multi-scale GIS. *International Journal of Geographical Information Systems*, 9(5), 489-507.

van Putten, J. & van Oosterom, P. (1998) New results with Generalized Area Partitionings. In In: Proceedings of the International Symposium on Spatial Data Handling, pp. 485-495. Vancouver, Canada.

Wood, J. (1996) The Geomorphological Characterisation of Digital Elevation Models. Doctoral Dissertation, University of Leicester, UK.

Wu, A. & Leahy, R. (1993) An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1101-1113.

Wu, H., Pan, M., Yao, L., & Luo, B. (2007) A partition-based serial algorithm for generating viewshed on massive DEMs. *International Journal of Geographical Information Science*, 21(9), 955-964.

Yang, W. & Gold, C.M.(1996) Managing spatial objects with the VMO-Tree. In Proceedings Seventh International Symposium on Spatial Data Handling (eds M.J. Kraak & M. Molenaar), pp. 711-726, Delft, The Netherlands.

Zhao, X., Chen, J., & Zhao, R. (1999) Dynamic Spatial Indexing Model Based on Voronoi. In Proceedings of the International Symposium on Digital Earth. Science Press