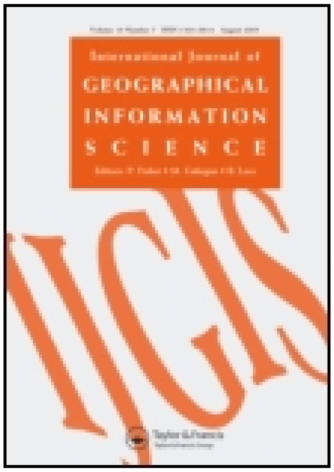

International Journal of Geographical Information Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis19

A comparative analysis of some two-dimensional orderings

DAVID J. ABEL [a] & DAVID M. MARK [a]

[a] Centre for Spatial Information Systems,CSIRO Division of Information Technology , GPO Box 664, Canberra, ACT 2601, Australia

Published online: 27 Apr 2007.

INT. J. GEOGRAPHICAL INFORMATION SYSTEMS, 1990, VOL. 4, NO. 1, 21–31

A comparative analysis of some two-dimensional orderings

DAVID J. ABEL and DAVID M. MARK†

Centre for Spatial Information Systems,
CSIRO Division of Information Technology, GPO Box 664, Canberra,
ACT 2601, Australia

Abstract. Many spatial analysis algorithms for cellular representations and data structures are based on an ordering of cells or objects to transform a two-dimensional problem to a one-dimensional one. Several orderings are available and their differing properties influence the performance of the data structures and algorithms. The relative merits of five orderings (row, row prime, Hilbert, Morton and Gray code) are assessed empirically for four paradigmatic geographical data-processing tasks in spatial analysis and data management. It is concluded that the Hilbert ordering deserves further investigation.

1. Introduction

The concept of ordering the cells of a gridded representation of a region is widely applied in geographical information systems (GIS) to reduce a two-dimensional problem to a one-dimensional problem to simplify algorithms and to take advantage of the extensive set of list data structures available (e.g., Knuth 1973, Aho *et al.* 1974, Wirth 1986). For such operations as overlaying and smoothing, it is usual to traverse the cells in a given order. The simplest and most common form is row order, where cells are manipulated in sequence by row and, within row, by column. For the design of spatial data structures, one strategy is to employ an ordering of objects as a list by deriving for each object a single-valued key from its coordinates. An example is the linear quadtree representation (Gargantini 1982, Samet 1984). Subquadrants are assigned Morton keys by bit-interleaving the coordinates of their origins. The list of subquadrants can then be implemented using any of a number of well-studied data structures and algorithms, such as a B-tree where the linear quadtree must be stored on disk.

Mark and Goodchild (1986) defined an ordering as a one-to-one reversible assignment of consecutive integers (or keys) to the members of a set of distinct spatial entities (polygons, cells or points). These keys can then be used to denote the sequence in which objects are traversed or as addresses within a list. Many orderings have been adopted or proposed (Goodchild and Grandfield 1983, Mark and Goodchild 1986, Mark, unpublished). Two orderings which are well-entrenched in GIS are row and Morton ordering (Morton 1966). Others which have recently been argued as deserving attention are row prime, Hilbert and Gray code.

It must be expected that the differing properties of the various orderings are reflected in differing degrees of suitability for various tasks. Consider, for example, a smoothing operation on a disk-resident image requiring, for each cell, access to its neighbours. An ordering which assigns similar keys to a cell and its neighbours, intuitively, will tend to minimize the number of disk accesses.

† Permanent address: National Center for Geographic Information and Analysis, State University of New York at Buffalo, New York 14260, U.S.A.

0269–3798/90 $3·00 © 1990 Taylor & Francis Ltd.

This paper reports empirical studies to explore the relative properties of five orderings (row, row prime, Morton, Hilbert and Gray code) by assessing their performances on four paradigmatic GIS usages. The first is representation of an object, considering run-encoded forms and adopting storage requirements as a measure of performance. The second is processing all cells of a region, with the manipulation of each cell requiring access to its rook's case neighbours. A paged memory environment is modelled with page faults used as a measure of performance. The third usage, addressing the use of ordering as the basis of data structures, is windowing point sets, with a point indexed by the cell within which it lies. Oversearch is adopted to measure performance. The final usage is evaluation of a point's nearest neighbour using an algorithm (Abel and Smith 1984 a) which relies on the correspondence between the keys for two points and their spatial proximity.

2. Two-dimensional orderings

Recently, Mark (1990) has identified three properties of a spatial ordering which are relevant to spatial data processing. These properties are defined for a discrete (cellular) model of space; they can be applied to points with continuous coordinates by overlaying a grid of cells sufficiently small that no cell contains more than one point.

(*a*) An ordering is *continuous* if, and only if, the cells in every pair with consecutive keys are rook's case (four-connected) neighbours (that is, they share a side); continuity is important for spatial movement and is desirable for run-encoding tasks.

(*b*) An ordering is *quadrant-recursive* if the cells in any valid quadtree subquadrant of the matrix are assigned a set of consecutive integers as keys. If the entire image is 2^n by 2^n cells, the image has n levels, and cells are of level 0. A valid subquadrant of level L is of sidelength 2^L and contains 2^{2L} cells; in a quadrant-recursive ordering, the cells will be ordered using some set of 2^{2L} consecutive integers. Quadrant-recursive orderings can be used as keys for linear quadtrees.

(*c*) An ordering is *monotonic* if, and only if, for every fixed x, the keys vary monotonically with y in some particular way (i.e., either always directly or always inversely), and likewise for every fixed y, the keys vary monotonically with x.

One of the best-known and most widely-used spatial orderings is most often referred to as Morton order (Morton 1966, Tomlinson *et al.* 1976). Morton keys can be formed simply by interleaving the bits of the x and y coordinates of each cell. Morton order is both monotonic and quadrant-recursive, and is used (at least implicitly) as the basis for most linear quadtree systems. At each step in the recursive construction of a Morton ordering, the subquadrants are numbered in a pattern in the shape of a letter 'N' or 'Z' so that alternative names for Morton order are 'Z order' (Orenstein 1982) or 'N order' (Goodchild and Grandfield 1983).

Hilbert ordering is based on the classic Peano curve. Its generating function was first published by Peano in 1890, but the first drawings of the curve were published by Hilbert in 1891 (see Kennedy 1973, p. 142). Since the term 'Peano curve' has been subsequently generalized to mean any continuous curve which fills a 2D or higher space in the limit, the term 'Hilbert order' is used here for this ordering. Hilbert order is both quadrant-recursive and continuous. It is therefore especially promising for spatial data-handling because it is quadtree-compatible and yet, because of its continuity, performs very well in run-encoding tests (cf. Goodchild and Grandfield 1983). It has been used in colour-space processing by Stevens *et al.* (1983), and was called 'Pi order' by Goodchild and Grandfield (1983) because of the shape of the level-1 components of the ordering.

Row order simply numbers the cells in a matrix in row-by-row order; each row is numbered in the same direction. Row order is the common ordering for raster data in remote sensing and image processing. It is monotonic and very simply implemented. A variant of row ordering is row prime ordering in which alternate rows are traversed in opposite directions (Goodchild and Grandfield 1983). It has also been referred to as 'snake-like' ordering, and is continuous but not monotonic.

Recently, Faloutsos (1987) suggested that keys based on Gray codes might be useful for spatial data handling. Briefly, Gray codes have the property that successive codes differ in exactly and only one bit position. Faloutsos suggested that spatial data could be organized by converting the x and y coordinates to Gray codes, interleaving the bits in those codes (as in the formation of a Morton key), and then transforming the resulting Gray code back to an order position. Such Gray-code keys are quadrant-recursive. Gray-code keys that differ by exactly one must either be in the same row or in the same column of the image but can be far apart.

3. Space requirements

Requirements of disk or RAM space have traditionally been considered important in assessing alternative forms of image representation and the data structures for chosen representations. Many image representations and data structures are not sensitive to the ordering of cells in the image. An uncompressed array representation, for example, has space requirements dependent only on the number of cells and the length of the intensity value for each. Similarly, a pointer-based or a linear quadtree representation has a space requirement dependent only on the number of leaves and of the black leaves respectively (Samet 1984). Run-encoded forms of image representations, however, are ordering-sensitive as they are based on representation of the image as sets of runs (i.e., by the subsets of cells contiguous in key sequence which have the same intensity values). The run-encoded form of an image is well known. The 2DRE linear quadtree (Mark and Lauzon 1984, 1985, Lauzon *et al.* 1985) employs run-encoding for cells using a Morton ordering. In these cases, differences in space requirements for the various orderings can be expected. Performance can here be measured as the number of subsets of consecutively-numbered runs (in key order) of cells which fall on the object.

Two data sets were used to test this aspect of performance. The first was a set of the minimum bounding rectangles (recorded as latitude and longitude) of the 1143 local government areas of Australia. Each rectangle was converted to an image representation by mapping its minimum bounding rectangle to a 256 by 256 integer coordinate system with simple integer truncation. This resulted in 236 of the rectangles being reduced to a single cell. These were not considered further, so that the following analyses are based on the remaining 907 rectangles. The second set was generated by forming all rectangles of integral side length whose x-ranges varied from 2 to 11 inclusive and whose y-ranges varied similarly, so that 100 rectangles were formed. Each of these rectangles was placed randomly 100 times on a 256 by 256 image under a uniform distribution so that in all 10 000 tests were conducted.

The results are given as tables 1 to 3 and indicate consistent relative performance of the five orderings for the two data sets. Row, row prime and Hilbert orderings are equivalent in performance in average number of runs. Hilbert ordering provides fewer

Table 1. Runs for local government areas for 5 orderings.

	Average number of runs Local government areas	Random rectangles
Morton	10·20	11·01
Hilbert	6·11	6·47
Gray code	10·09	10·70
Row	6·18	6·49
Row prime	6·16	6·46

Table 2. Cases where run counts differ for orderings (local government areas).

	Number of cases				
Ordering	Morton	Hilbert	Gray code	Row	Row prime
Morton	—	694	98	645	646
Hilbert	24	—	26	345	348
Gray code	0	669	—	629	630
Row	94	322	115	—	8
Row prime	90	317	111	0	—

Values down a column are counts of cases where the ordering at the head of the column required fewer runs than the ordering for the row. As examples, Morton ordering had fewer runs than Hilbert in 24 cases, but Hilbert had fewer runs than Morton in 694 cases. In the balance of 189 cases, the two orderings required identical numbers of runs.

Table 3. Cases where run counts differ for orderings (random rectangles).

	Number of cases				
Ordering	Morton	Hilbert	Gray code	Row	Row prime
Morton	—	9333	2171	8446	8465
Hilbert	55	—	233	4196	4222
Gray code	0	9027	—	8194	8214
Row	1084	4264	1224	—	78
Row prime	1060	4227	1200	0	—

Values down a column are counts of cases where the ordering at the head of the column required fewer runs than the ordering for the row. As examples, Morton ordering had fewer runs than Hilbert in 55 cases, but Hilbert had fewer runs than Morton in 9333 cases. In the remaining 612 cases, the two orderings required identical numbers of runs.

runs than row or row prime in approximately 30 per cent of the cases but more runs for 30 per cent. The data for Morton and Gray-code orderings confirm the analysis of Faloutsos (1987), showing that Gray code is at worst equal to bit-interleaved ordering, a modified form of Morton ordering. However, the number of cases where Gray-code ordering provides fewer runs than Morton is small (10 per cent for local government areas and 20 per cent for the random rectangles) and the differences in average runs is slight. Morton and Gray-code ordering both perform poorly relative to the other orderings.

4. Image analysis

Many procedures of spatial analysis require the examination of the local neighbourhood of a given cell. Consider, for example, the smoothing of an image by replacing a cell's intensity by the weighted average of the intensities of the cell and its rook's case neighbours, or the calculations for slope, aspect and hill-shading in a digital elevation model. Clearly the costs of accessing neighbouring cells are dependent on the ordering used and it is relevant to adopt the access costs as a criterion of merit in assessing the orderings. Where an image can be held wholly in RAM, all cells are available with low access costs and the only difference between orderings is due to the relative ease of computing the linear keys for neighbouring cells. Where a disk must be used, either directly for storage of the image or indirectly through use of virtual memory, the disk accesses incurred are a more influential component of the total costs of the analysis.

The first evaluation of support for spatial analysis adopted examination of the four rook's-case neighbours for all cells of an image in key sequence as an elemental operation, with virtual memory page faults adopted as the measure of costs of the operation. The (square) image was assumed to be segmented into pages of a fixed size B, measured in cells. (For example, if the image was a Landsat Thematic Mapper image with 7 bytes for cell intensity and the disk physical blocksize was 512 bytes, the blocksize measured under the model would be 73 cells.) The 0th page then held intensities for the cells with key values 0 to (B–1), the second B to (2B–1) and so on. Physical memory was considered as 5, 9, 18 or 27 pages, with page management following an 'oldest-last-access' discipline to minimize page swapping (that is, when a page swap from physical to virtual memory is required, the page in physical memory to be replaced was the page with the greatest time since the last access to it). A constant cost to fetch a page from disk to RAM was assumed.

A range of images whose sidelengths were integral multiples of B was tested for Morton, Hilbert, Gray code, row and row-prime orderings. The average accesses per page are reported in table 4. Note that the tests exclude the important case, for row and row prime, where B is greater than, or equal to, the sidelength of the image. Here simple data management (maintaining in physical memory the equivalent of three rows of the image) allows each page to be fetched once only, so that each page is fetched from disk once only.

As expected, the results show that the number of page accesses from disk is dependent on the size of the cache and the size of the image. For small caches (where only 5 pages can be held in RAM), except for the smallest image size tested, Hilbert is better than only Gray-code ordering. For larger caches, however, Hilbert leads to fewer page accesses from disk. Morton ordering is better, for all cache and image sizes, than row and row prime orderings. The performance of Gray-code ordering lies between that of the Morton and of the row and row-prime orderings for larger cache sizes.

5. Spatial search

A significant use of 2D to 1D ordering is in spatial search operations, of which windowing is perhaps the most common GIS operation in its own right (for example, to retrieve objects in a certain subregion for display) and to fetch a set of candidate objects to be subjected to further tests (e.g., Abel 1986). We consider here the case where objects are points and the window is rectangular; conceptually similar approaches for extended objects are described by Abel and Smith (1983, 1984 b). The general strategy is

Table 4. Page accesses for neighbour access.

Cache size (number of blocks)	Image size (number of blocks) 16	64	256	1024	4096	16384
	Average number of page accesses for neighbour access					
Morton						
5	2·31	2·89	3·22	3·35	3·42	3·46
9	1·50	2·14	2·51	2·70	2·79	2·83
18	1·00	1·36	1·70	1·86	1·93	1·97
27	1·00	1·06	1·51	1·65	1·73	1·77
36	1·00	1·00	1·28	1·46	1·57	1·63
Hilbert						
5	2·25	3·05	3·51	3·74	3·88	3·93
9	1·25	1·92	2·31	2·52	2·64	2·69
18	1·00	1·47	1·68	1·82	1·89	1·92
27	1·00	1·13	1·40	1·56	1·64	1·69
36	1·00	1·13	1·29	1·39	1·45	1·48
Gray code						
5	2·75	3·44	3·74	3·88	3·94	3·97
9	1·75	2·41	2·82	3·00	3·10	3·14
18	1·00	1·69	1·98	2·12	2·19	2·22
27	1·00	1·38	1·65	1·82	1·91	1·95
36	1·00	1·13	1·48	1·64	1·72	1·76
Row						
5	2·50	2·75	2·88	2·94	2·97	2·98
9	1·94	2·75	2·88	2·94	2·97	2·98
18	1·00	2·56	2·88	2·94	2·97	2·98
27	1·00	1·00	2·88	2·94	2·97	2·98
36	1·00	1·00	2·80	2·94	2·97	2·98
Row prime						
5	2·19	2·83	3·16	3·30	3·41	3·46
9	1·44	2·17	2·57	2·78	2·89	2·95
18	1·00	1·67	2·32	2·66	2·83	2·91
27	1·00	1·16	2·02	2·50	2·75	2·88
36	1·00	1·00	1·77	2·38	2·69	2·84

organization of a set of objects as a list using, as the key value (under a chosen ordering) for a point, the key of the covering cell. To enable fast access by ranges of key values, the list is implemented as an ISAM (indexed, sequential access method) data structure such as a binary tree (for RAM) or a B-tree (for disk). The search by a window can then proceed by searching the list for those points whose key values match cells covered by the window. A conceptually similar approach is to determine the runs of cells (i.e., the lowest and highest key values for a run) covered by the window and search by these ranges. While this has zero oversearch (i.e., no points outside the rectangle are fetched), typically many direct accesses by key value (as many as there are runs) are incurred. The other extreme is to retrieve all points whose keys lie in the range from the minimum-key-value cell within the window to the maximum-key-value. Clearly this requires a single random access, but typically will incur a high oversearch. An intermediate

Table 5. Oversearch for rectangle data sets.

	Morton	Hilbert	Gray code
Average oversearch for rectangle data sets as ratio of key range and area of rectangle			
1143 Australia local government areas			
Minimum covering quadrant	63·61	63·61	63·61
Total key range	20·26	22·75	29·52
Two/four covering quadrants	3·11	3·11	3·11
Two/four key ranges	1·84	1·93	2·11
10000 randomly-placed small rectangles			
Minimum covering quadrant	128·05	128·05	128·05
Total key range	35·38	42·49	51·78
Two/four covering quadrants	2·97	2·97	2·97
Two/four key ranges	1·79	1·83	1·99

approach is to choose a few ranges, such that the ranges used include relatively few values outside the window. A suitable heuristic is to choose the ranges corresponding to the minimum- and maximum-key values for the portions of the window formed by splitting the window along the major subquadrant boundaries within it (Abel and Smith 1984 b).

Clearly the various orderings will lead to differing oversearch where the search is executed by key ranges, as an object, under the various orderings, will have differing key ranges. To assess the extent of the differences, the ratio of the key ranges to the area of objects was evaluated for Morton, Hilbert and Gray-code orderings for the local government areas and the 10 000 random rectangles. (Row and row-prime orderings were not assessed, as they are clearly poorly-suited to the algorithm modelled.) The forms of search modelled were search by a single key range and by the ranges obtained by splitting along major subquadrant boundaries. The results are given in table 5.

In this case Morton ordering provides the lowest oversearch, with Hilbert better than Gray code. For both test sets with searching by a single key range, Hilbert is higher than Morton by approximately 20 per cent while Gray code is higher by 50 per cent. Significant reductions in oversearch are achieved by splitting the key range—here Morton and Hilbert provide very similar oversearch, with Gray code now approximately 10 per cent higher. The search algorithms are modifications of those of Abel and Smith (1983, 1984 b) for sets of extended objects. There the window was searched in terms of the key range for its minimum covering subquadrant or the key ranges of the four smallest descendants which together covered the window. Table 5 also includes the oversearch for these disciplines for the search of points by rectangular windows. Clearly the use of key ranges is to be preferred for point sets.

6. Nearest neighbour search

The nearest-neighbour problem deals with the identification of the point from a set of points which is closest according to some distance metric (we assume Euclidean distance here) to a specified query point. The nearest-neighbour problem is frequently encountered within spatial analysis algorithms (Hartigan 1975, Davis and McCullagh 1975) and its own right. It has been shown (Abel and Smith 1984 a) to be efficiently executed (in terms of expected-case performance) by exploiting properties of Morton ordering. We describe here a variation of Abel and Smith's algorithm and report the relative performances of Morton and Hilbert code ordering. The algorithm is especially relevant here as it relies on the preservation of spatial contiguity in key space.

The technique is based on organizing the set of points as a list in ascending sequence by their spatial keys. The algorithm has two steps. Firstly, the spatial key for the query point is evaluated and the points of the set whose keys are immediately above and below the key of the query point are retrieved. The distances from these points to the query point are determined, with the minimum adopted as an estimate of the distance to the nearest neighbour. (Use of two points guards against the query point lying near the corner of a major subquadrant.) Then all points within a square region with a side length of twice the estimate, centred on the query point, are retrieved as a windowing problem. The distances to the query point from all these points are evaluated to identify the nearest neighbour to the query point.

Clearly the effectiveness of the algorithm requires that points with similar keys are also close in Euclidean space. The choice of ordering can then be expected to influence the performance of this algorithm through the influence on the estimate of the distance to the nearest neighbour and, as considered above, through the effect on oversearch within the search of the points by a rectangular window. Performance was measured as the number of points within the search region (i.e., the square region of side length twice the estimate of the distance to the nearest neighbour). The tests conducted considered only Morton and Hilbert keys on search of sets of 256, 1024, 4096 and 16 384 points. All tests adopted a fixed grid of 256 by 256 cells.

The average counts of points examined (table 6) show that the Hilbert ordering is superior. This is due to Hilbert ordering providing better initial candidates. In the previous section, Morton was shown to provide greater efficiency in windowing. If the estimated distance to the nearest neighbour is lower by some ratio $r < 1$, then the area of the retrieval rectangle will be smaller by a ratio of r^2. If the density of points is uniform, then the number of points in that rectangle should also reduce by a ratio of r^2, which is approximately the observed result. The differences, however, are not great and in general amount, on average, to a single point examined.

At first inspection, the results appear not to be in accord with the conclusion of Abel and Smith (1984 a) that the expected numbers of points examined is independent of the number of points in the database. The analysis of Abel and Smith, however, assumed that the average number of points assigned the same key (i.e., in the same cell) is very small, a condition not present in the tests here. Additionally, the data reported here appear to show effects of a relatively coarse subdivision (a grid of 256 by 256) for large numbers of points, with the proportional oversearch increasing as the estimate of the distance to the nearest neighbour approaches the size of the cells. In more detail, the increase in points retrieved can be attributed to the fact that the set of cells covering a rectangle has a combined area at least somewhat larger than the rectangle itself. For the Morton keys, the average sidelength of the search window for 256 points is 21·16, which, in most cases, would be covered by a square subregion of 22 by 22 cells. The ratio of the area of the subregion to the search window is then 1·08. For 16 384 data points, the search window has an average sidelength of 3·02. However, such a window would include parts of, at best, a 4 by 4 cell (area = 16) subregion and, for some placements, a 4 by 5 or even 5 by 5 subregion. With the lower figure, the area of the discretized rectangle is expected to be 1·75 times the area of the actual rectangle. The ratio of these (1·62) is close to the ratio of the average points examined for sets of 256 and 16 384 points (1·79).

Table 6. Comparison of Morton and Hilbert keys for nearest neighbour search.

	Morton	Hilbert	Ratio
256 points in the database			
Average number of points examined	4·17	3·88	0·93
Mean distance to initial candidate	10·58	10·21	0·97
Mean distance to nearest neighbour	7·99	8·05	—
Mean ratio of candidate to minimum distance	1·43	1·33	0·92
1024 points in the database			
Average number of points examined	5·22	4·52	0·87
Mean distance to initial candidate	5·50	5·13	0·93
Mean distance to nearest neighbour	4·15	4·15	—
Mean ratio of candidate to minimum distance	1·40	1·29	0·92
4096 points in the database			
Average number of points examined	5·80	4·91	0·85
Mean distance to initial candidate	2·85	2·57	0·90
Mean distance to nearest neighbour	2·03	2·03	—
Mean ratio of candidate to minimum distance	1·60	1·40	0·88
16384 points in the database			
Average number of points examined	7·46	6·24	0·84
Mean distance to initial candidate	1·51	1·37	0·91
Mean distance to nearest neighbour	1·01	1·01	—
Mean ratio of candidate to minimum distance	1·79	1·56	0·87

Note: Points were distributed over a 256×256 unit square, and all statistics reported are averages for 1024 randomly-distributed query points.

7. Conclusions

The performance of five orderings for a set of representative GIS operations have now been reported. The assessment has not included three aspects which, for some combinations of ordering and operations, significantly influence the total costs of executing a set of operations on a data set. The first is the cost of evaluating a key of a given cell and of the inverse operation. The second is the need, in some cases, to re-order an image, such as the transformation of an image in row order as acquired to Morton ordering for manipulation and analysis. The third is the tailoring of an algorithm to exploit strengths or to ameliorate weaknesses of an ordering. To this extent the data presented should be treated as suggestive of the general utility of the orderings and of profitable avenues for further research and development, rather than as enabling a definitive statement of relative merit. The row and Morton orderings are both well-entrenched in certain GIS sectors and can be treated as benchmarks to assess the row-prime, Hilbert and Gray-code orderings which are yet to be widely adopted.

Row-prime ordering, a relatively minor variant of row ordering, has been shown to provide only slightly improved performance in terms of space requirements under run-encoded representations. It does provide a better performance in terms of disk accesses for operations involving access to neighbouring cells, except in the cases of a large image and a small cache. However, the test excluded the case where at least three rows of the image could be held in RAM. In this case a simple approach removes the overheads in disk access for row ordering. On this basis row-prime ordering appears to have no significant merit. Similarly there are no grounds to advocate the use of Gray-code ordering as a replacement for Morton ordering. Gray code provides only slight improvements in space requirements for run-encoded representations and has higher oversearch for windowing operations.

The tests suggest that Hilbert ordering deserves closer attention as an alternative to Morton ordering for many operations. It is more effective as a basis for run-encoded representations (by approximately 40 per cent) and, except for large images and small cache sizes, for neighbour-access operations. It is poorer than Morton ordering in terms of oversearch for windowing operations but better (as assessed by evaluation of nearest neighbours) in preserving spatial contiguity. The major weakness remains the lack of inexpensive algorithms to evaluate the key of a cell or subquadrant from its coordinates and vice versa.

Acknowledgments

This research was conducted at the Centre for Spatial Information Systems of the CSIRO Division of Information Technology, Canberra, Australia, while Mark was a Visiting Scientist. Mark was also partly supported by U.S. National Science Foundation Grant SES-86-18778. The support of the Foundation and of CSIRO is gratefully acknowledged. Dr J. L. Smith and Dr J. F. O'Callaghan are thanked for useful comments on earlier forms of this paper.

References

Abel, D. J., 1986, Bit-interleaved keys as the basis for spatial access in a front-end spatial database management system. *Proceedings of the Tesseral Workshop No. 2 held in Reading on 22–23 September 1986*, edited by B. Diaz and S. B. M. Bell (Swindon: Natural Environment Research Council), pp. 163–173.

Abel, D. J., and Smith, J. L., 1983, A data structure and algorithm based on a linear key for rectangle retrieval problems. *Computer Vision, Graphics, and Image Processing*, **24**, 1.

Abel, D. J., and Smith, J. L., 1984 a, A simple approach to the nearest neighbour problem. *The Australian Computer Journal*, **16**, 140.

Abel, D. J., and Smith, J. L., 1984 b, A data structure and retrieval algorithm for a database of areal entities. *The Australian Computer Journal*, **16**, 147.

Aho, A. V., Hopcroft, J. E., and Ullman, J. D., 1974, *The Design and Analysis of Computer Algorithms* (Massachusetts: Addison-Wesley).

Davis, J. C., and McCullagh, M. J. (editors), 1975, *Display and Analysis of Spatial Data* (New York: John Wiley).

Faloutsos, C., 1987, Gray codes for partial match and range queries. Technical Report UIAMCS-TR-87-4, CS-TR-1796, University of Maryland.

Gargantini, I., 1982, An effective way to represent quadtrees. *Communications of the ACM*, **25**, 905.

Goodchild, M. F., and Grandfield, A. W., 1983, Optimizing raster storage: an examination of four alternatives. *Proceedings of the Auto-Carto 6 Conference held in Canada on 16–21 October 1983*, edited by B. S. Wellar (Ottawa: the Steering Committee for the Sixth International Symposium on Automated Cartography), pp. 400–407.

Hartigan, J. A., 1975, *Clustering Algorithms* (New York: John Wiley).

Kennedy, H. C., 1973, *Selected Works of Giuseppe Peano* (translated and edited by H. C. Kennedy (London: George Allen & Unwin Ltd).

Klinger, A., and Dyer, C., 1976, Experiments on picture representation using regular decomposition. *Computer Graphics and Image Processing*, **5**, 68.

Knuth, D. E., 1973, *The Art of Computer Programming*, Vol. 3 *Sorting and Searching* (Massachusetts: Addison-Wesley).

Lauzon, J. P., Mark, D. M., Kikuchi, L., and Guevara, J. A., 1985, Two-dimensional run-encoding for quadtree representation. *Computer Vision, Graphics and Image Processing*, **30**, 56.

MARK, D. M., 1990, Neighbour-based properties of some orderings of two-dimensional space. *Geographical Analysis* (in the press).

MARK, D. M., and GOODCHILD, M. F., 1986, On the ordering of two-dimensional space: introduction and relation to Tesseral principles. *Proceedings of the Tesseral Workshop No. 2 held in Reading on 22–23 September 1986*, edited by B. Diaz and S. B. M. Bell (Swindon: Natural Environment Research Council), pp. 179–192.

MARK, D. M., and LAUZON, J. P., 1984, Linear quadtrees for geographic information systems. *Proceedings of the International Symposium on Spatial Data Handling held in Zurich in 1984*, edited by D. F. Marble, K. E. Brassel, D. J. Peuquet and H. Kishimoto (Zurich: Department of Geography, University of Zurich-Irchel), pp. 412–430.

MARK, D. M., and LAUZON, J. P., 1985, The space-efficiency of quadtrees: an empirical examination including the effects of two-dimensional run-encoding. *Geo-Processing*, **2**, 367.

MORTON, G. M., 1966, A computer oriented geodetic data base, and a new technique in file structuring. Unpublished report, IBM Canada Ltd. See also Tomlinson *et al.* (1976).

ORENSTEIN, J. A., 1982, Algorithm and data structures for the implementation of a relational database system. Ph.D. Dissertation, School of Computer Science, McGill University, Montreal, Canada.

SAMET, H., 1984, The quadtree and related hiearchical data structures. *Association for Computing Machinery Surveys*, **16**, 187.

STEVENS, R. J., LEHAR, A. F., and PATTERSON, F. H., 1983, Manipulation and presentation of multidimensional image data using the Peano scan. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, **5**, 520.

TOMLINSON, R., CALKINS, H. W., and MARBLE, D. F., 1976, *Computer Handling of Geographic Data: An Examination of Selected Geographic Information Systems* (Paris: UNESCO Press).

WIRTH, N., 1986, *Algorithms and Data Structures* (London: Prentice-Hall International).