ELSEVIER

Computers & Geosciences 31 (2005) 1042–1047

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

GHAM: A compact global geocode suitable for sorting ☆

Duncan Carr Agnew*

Institute of Geophysics and Planetary Physics, Scripps Institution of Oceanography, University of California, San Diego, La Jolla CA 92093-0225, USA

Received 2 February 2004; received in revised form 25 February 2005; accepted 25 February 2005

Abstract

The GHAM code is a technique for labeling geographic locations based on their positions. It defines addresses for equal-area cells bounded by constant latitude and longitude, with arbitrarily fine precision. The cell codes are defined by applying Morton ordering to a recursive division into a 16 by 16 grid, with the resulting numbers encoded into letter–number pairs. A lexical sort of lists of points so labeled will bring near neighbors (usually) close together; tests on a variety of global datasets show that in most cases the actual closest point is adjacent in the list 50% of the time, and within 5 entries 80% of the time.

Keywords: Geocode; Global grid; Morton ordering

1. Introduction

A perennial problem with geographic data is describing their location in a compact and useful form; as Merrill (1986) pointed out, many of the locations given in the geological literature are either imprecise or in special systems of limited use. While geographical coordinates are sufficient, and readily available using GPS, much experience has shown that they are not always easy to work with—which has led most mapping authorities to provide alternative grid systems for locating points. Globally, the most familiar of these is the UTM grid. From the standpoint of managing geographic data, a defect of both geographic and grid coordinates is that they do not always provide an address for a location that will indicate which other locations are nearby and which farther off. In part this is an unavoidable byproduct of the space in which such data are located: in more than one dimension there is no natural order for sorting information.

A large number of location-based identifiers have been developed—indeed, the number itself indicates how perennial this problem is. Most of the better-known ones (such as the various national grids) apply to only a small part of the Earth's surface; those that are global mostly extend to only a limited level of detail that can be insufficient for many datasets. Some rely on geographic divisions not easy to describe compactly (such as political boundaries), which makes them cumbersome to compute. And others (such as telephone area codes) are explicitly designed to make nearby codes dissimilar: an advantage in avoiding confusion by users, but the opposite of what is wanted if a list is to be sorted.

In the course of developing a database of geodetic monuments for the Southern California Earthquake Center, it became clear that there was considerable value in being able to assign an address to each location, and do so in such a way that a sorted list of the addressed

☆ *URL*: http://www.iamg.org/CGEditor/index.htm

*Tel.: +1 8585342590; fax: +1 8585345332.
E-mail address: dagnew@ucsd.edu.

0098-3004/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2005.02.007

points would have the ones that are nearby in space and also be close together in the list. I have developed a code to meet this need that is easily computed, compact, globally applicable, and extensible to arbitrary detail, and which gives addresses with the desired sorting properties. The code described here, a slight modification of the original one, is called the GHAM code for Global, Hierarchical, Alphanumeric, and Morton-encoded: all terms that will be more obvious once the code is explained.

2. Description of the code

While many systems for dividing up the Earth start with projections onto Platonic solids (Sahr et al., 2003), followed by subdivision into grids of triangles (e.g., Goodchild and Yang, 1992; Dutton, 1998) it is easier to visualize (and compute) a subdivision based on latitude–longitude rectangles, as was done by Tobler and Chen (1986), whose methods I in part follow. In order to make the final cells all have the same area, we begin with an equal-area projection of the surface of the sphere (or, to a perfectly adequate approximation in this case, of the ellipsoid). We convert the geographic latitude ϕ and longitude λ into x and y coordinates using an equal-area cylindrical projection:

$$x = (\lambda + 180.)/360., \quad y = 0.25[\sin(\phi) + 1.], \tag{1}$$

where λ is given in degrees, with $-180^\circ \leqslant \lambda \leqslant 180^\circ$. This scaling means that x runs from 0 to 1, and y from 0 to 0.5, giving a map with the overall aspect ratio shown in Fig. 1 (note that this map is not itself the equal-area projection).

The next steps are done as many times as needed to get the necessary level of subdivision:

(1) Multiply x and y by 16 to get numbers r_x and r_y; take the integer parts i_x and i_y of these. These integers run from 0 to 15, and so can be represented with 4 bits.
(2) Interleave the bits of i_x and i_y to form an 8-bit integer i_c. This is known as Morton ordering, after G. Morton, who developed it in 1966 for exactly the purpose of sorting geographic data—in his case, magnetic tapes of such data (Tobler and Chen, 1986; Mark, 1990). The value of i_c will be between 0 and 255 inclusive.
(3) Express i_c as $10n + m$, with n running from 0 through 25 and m from 0 through 9; then encode this value as a letter-numeral pair, using the $n + 1$th letter of the Latin alphabet, and the numerals from 0 through 9, so that A0 encodes the value 0, and Z5 the value 255.
(4) Define a new x and y by

$$x = r_x - i_x, \quad y = r_y - i_y,$$

which gives x and y between 0 and 1, and return to step 1. Continue for as many levels as desired.

All of this is simple to do in any programming language; Fortran and C programs are provided on the IAMG

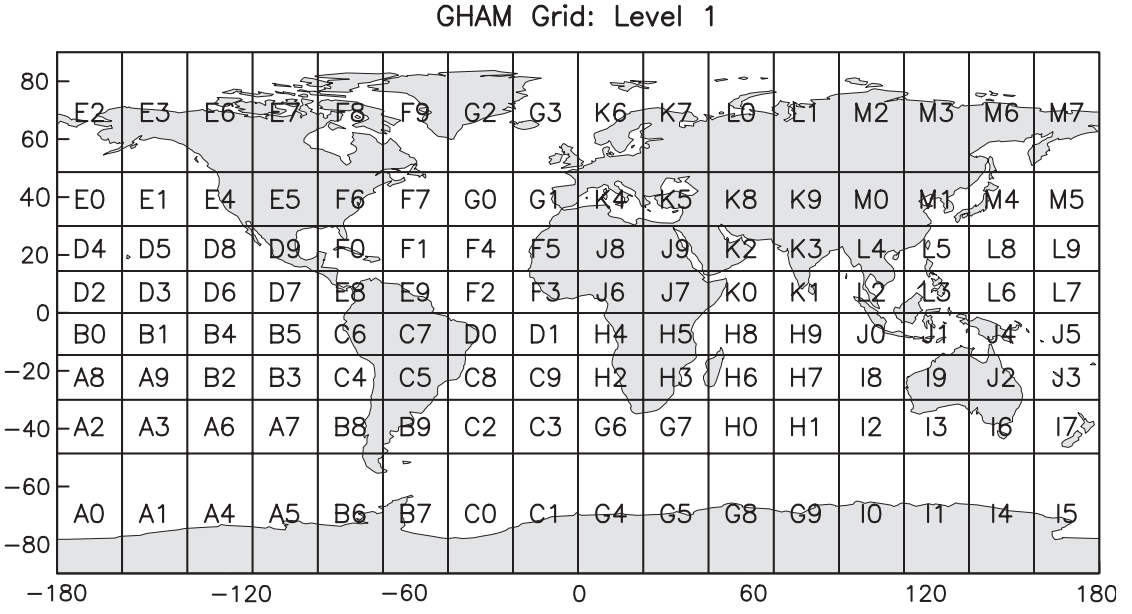

Fig. 1. Map showing division of globe into level-1 GHAM codes.

website. In fact, it is easy to perform the algorithm with only a hand calculator, given that steps 2 and 3 can be described once and for all through a table of codes for all possible values of i_x and i_y. Fig. 2 shows such a table, which also illustrates how a lexical sort of the codes will give a path which usually moves between nearby cells, with occasional large jumps. Appendix A shows an example of such a calculation.

The initial scaling in Eq. (1) is designed to make the first level fit within one part (the lower half) of the code table in Fig. 2, to give twice as many cells East–West as North–South. Since the total distance on the Earth in these two directions is in the same proportion, this scaling produces cells that are relatively compact over a wide range of latitudes. At the equator the aspect ratio (height/width) of cells in ground distance approaches $2/\pi$ as the level increases. This aspect ratio decreases to one at latitudes $\pm 37.07^\circ$, and reaches $\pi/2$ at $\pm 50.46^\circ$. Therefore, about 80% of the Earth's surface has cells with an aspect ratio of less than 1.6, though (unavoidably), the cells are very far from square close to the poles.

Those familiar with spatial databases will recognize Fig. 2 as an extended version of the quadtree index, in which data are labeled by recursive application of 2×2 arrays. Fig. 2 effectively does this for four levels of quadtree indexing at once, using Morton ordering to preserve proximity and the (purely chance) fact that there are 26 letters in the Latin alphabet to compactly encode all the possible combinations. The consequence of this is that each additional level in the GHAM code represents a 256-fold decrease in the area of the cells. Applying the algorithm four times, to create an 8-character code, corresponds to a cell size (measured by the square root of the area) of 480 m; going to Level 6 (a 12-character code) gives a cell size of 1.9 m; and to level 7 a size of 0.12 m. Given a restriction to letter–number pairs, Fig. 2 shows that almost all the possible codes are used, so that the GHAM code is very efficient (except for using only half the codes at the first level). At level 6, for example, 93% of all the codes that are possible are actually used. At the same time, fixing the code form to alternating letters and numbers provides some structure to guard against blunders, as well as avoiding the possible confusions between 1 and I or 0 and O.

The global nature of the GHAM code actually makes it especially convenient for addressing locations within smaller regions; because the codes are unique to a region, they may be assigned by different groups working in different areas with no fear of overlapping in the addresses used. Such overlap has been a real difficulty for the codes used, for example, for seismic and geodetic stations, requiring a central authority to adjudicate disputes. A location-based global code automatically avoids this problem.

Morton—ordered Alphanumeric Codes

i_y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	R0	R1	R4	R5	S6	S7	T0	T1	X4	X5	X8	X9	Z0	Z1	Z4	Z5
14	Q8	Q9	R2	R3	S4	S5	S8	S9	X2	X3	X6	X7	Y8	Y9	Z2	Z3
13	Q2	Q3	Q6	Q7	R8	R9	S2	S3	W6	W7	X0	X1	Y2	Y3	Y6	Y7
12	Q0	Q1	Q4	Q5	R6	R7	S0	S1	W4	W5	W8	W9	Y0	Y1	Y4	Y5
11	N8	N9	O2	O3	P4	P5	P8	P9	U2	U3	U6	U7	V8	V9	W2	W3
10	N6	N7	O0	O1	P2	P3	P6	P7	U0	U1	U4	U5	V6	V7	W0	W1
9	N0	N1	N4	N5	O6	O7	P0	P1	T4	T5	T8	T9	V0	V1	V4	V5
8	M8	M9	N2	N3	O4	O5	O8	O9	T2	T3	T6	T7	U8	U9	V2	V3
7	E2	E3	E6	E7	F8	F9	G2	G3	K6	K7	L0	L1	M2	M3	M6	M7
6	E0	E1	E4	E5	F6	F7	G0	G1	K4	K5	K8	K9	M0	M1	M4	M5
5	D4	D5	D8	D9	F0	F1	F4	F5	J8	J9	K2	K3	L4	L5	L8	L9
4	D2	D3	D6	D7	E8	E9	F2	F3	J6	J7	K0	K1	L2	L3	L6	L7
3	B0	B1	B4	B5	C6	C7	D0	D1	H4	H5	H8	H9	J0	J1	J4	J5
2	A8	A9	B2	B3	C4	C5	C8	C9	H2	H3	H6	H7	I8	I9	J2	J3
1	A2	A3	A6	A7	B8	B9	C2	C3	G6	G7	H0	H1	I2	I3	I6	I7
0	A0	A1	A4	A5	B6	B7	C0	C1	G4	G5	G8	G9	I0	I1	I4	I5

i_x

Fig. 2. Layout of GHAM codes for all levels greater than 1.

3. GHAM sorting and nearest neighbors: a comparison

As noted above, a major reason for developing GHAM was that a GHAM-coded list would, after sorting, group points nearby in a sorted list that would be nearby in space. Of course, this cannot be so in general, since the only sorting that would satisfy this condition (a sort by distance from a central point) would be different for each central point. But we may hope that a spatial sort on a code such as GHAM will, on average, provide a better level of grouping than other methods.

To test this I examined one artificial and four real distributions of global points. For each distribution I created a list in which GHAM codes (to Level 6) had been assigned to each point and the list sorted on these codes. I also found, for each point in the list, its nearest neighbor, and looked at the "list-distance": that is, the number of entries in the sorted list between these two points. One measure of performance is the fraction of points for which the list-distance is less than some number N. I look specifically at $N = 1$ (meaning that the points were adjacent in the list), $N = 2$ (the points were separated by no more than one entry between them) and $N = 5$. Dutton (1998) used as a criterion the distribution of geographical distances between points in a sorted list, compared with the shortest and longest path among all the points, but this was not a feasible criterion for the much larger numbers of points I used.

The one artificial point set was points distributed randomly over the surface of the Earth, easily computed by producing uniform random numbers to give the initial x and y coordinates on an equal-area projection. The other point sets were empirical, chosen to show different forms of the clustering that is common in spatial data. These datasets, shown in Fig. 3, are:

(1) Cities with population of 200,000 or more, taken from the World Cities Population Database, compiled for the United Nations Environment Program (http://www.grid.unep.ch/data/grid/gnv29.php). This list has 1157 locations, mostly in China, India, Europe, and the eastern United States.
(2) Permanent GPS systems, that are or have been used to monitor these satellites and measure crustal motion, taken from a list compiled by the Scripps Orbit and Permanent Array Center (http://sopac.ucsd.edu). This list has 1220 entries, some of them separated by only a few meters (multiple systems at one site); these are concentrated in the United States and Europe, with fairly uniform spacing elsewhere. (The very large Japanese GPS network is not in this list.)
(3) Earthquakes between 1900 and 1999 with magnitude 7 or above: 1608 events, taken from the centennial earthquake catalog of the International Association of the Physics of the Earth's Interior (Engdahl and Villasenor, 2002; http://earthquake.usgs.gov/scitech/centennial.html). Their locations are concentrated along the major plate

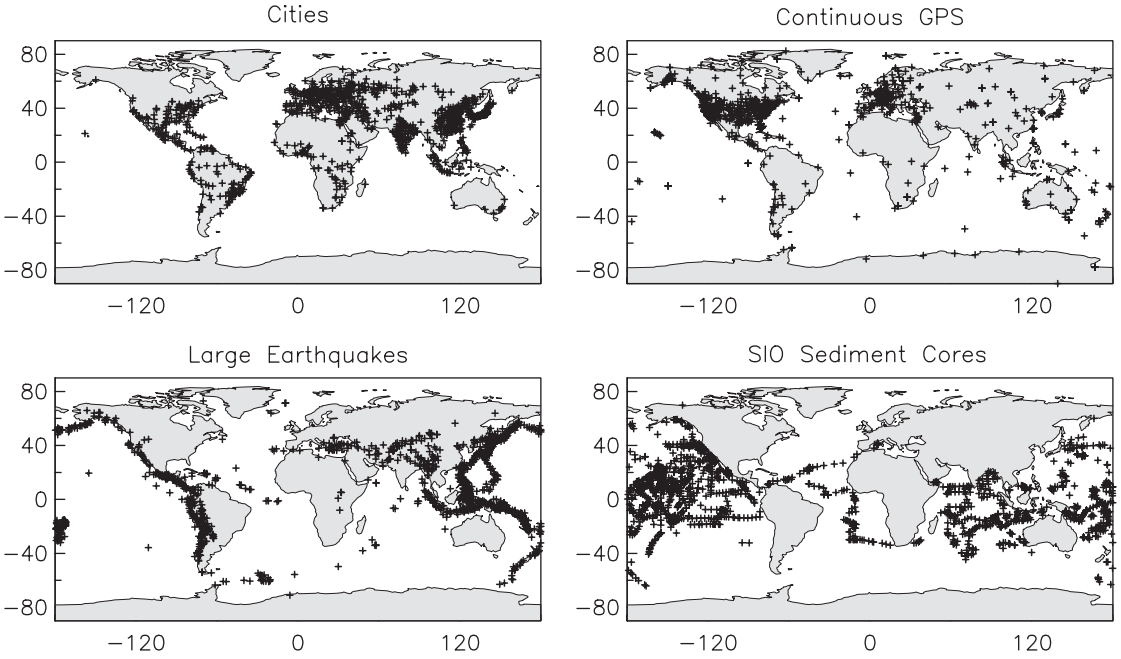

Fig. 3. Datasets used to test GHAM code and other global addressing algorithms.

boundaries, especially the subduction zones around the Pacific Ocean.

(4) Locations of sediment cores taken by the Scripps Institution of Oceanography between 1960 and 1970, taken from the curator's index of oceanic and lacustrine samples kept by the National Geophysical Data Center. (http://www.ngdc.noaa.gov/mgg/curator/curator.html) These 1627 locations are strongly lineated because they are along the tracks of the ships from which they were made.

Table 1
Proximity of nearest neighbors in sorted lists

Dataset	Sort type	$N=1$	$N\leqslant 2$	$N\leqslant 5$
Cities	GHAM	52	66	80
	G and Y	36	50	71
	Dutton	22	29	41
	B and G (Quat)	26	34	45
	B and G (Bin)	49	63	79
	Longitude	11	19	35
	Latitude	31	49	74
	Alphabetical	2	3	4
GPS Stations	GHAM	59	69	81
	G and Y	53	62	74
	Dutton	51	61	74
	B and G (Quat)	59	70	78
	B and G (Bin)	59	70	79
	Latitude	31	39	54
	Longitude	36	45	62
	Alphabetical	22	23	25
Earthquakes	GHAM	59	70	81
	G and Y	43	54	68
	Dutton	43	54	68
	B and G (Quat)	52	62	75
	B and G (Bin)	50	60	74
	Longitude	22	33	53
	Latitude	30	46	70
	Chronological	4	6	7
Cores	GHAM	60	69	78
	G and Y	47	58	71
	Dutton	47	59	71
	B and G (Quat)	55	65	75
	B and G (Bin)	54	63	74
	Longitude	24	34	52
	Latitude	26	39	60
	Chronological	49	57	63
Random	GHAM	30	37	42
	G and Y	22	29	35
	Dutton	23	30	36
	B and G (Quat)	26	32	39
	B and G (Bin)	27	33	39
	Latitude	5	11	23
	Longitude	4	7	19

Table 1 gives the results for these datasets, in the case of the randomly-distributed points the median of 10 trials of 1000 points each. Each value is the percentage of entries for which the actual nearest neighbor can be found within N positions in the list, as sorted in the following ways: by GHAM code, by four codes based on triangular subdivisions of the Earth's surface, by latitude, by longitude, and by some "natural" identifier, such as alphabetically by name or chronologically by date. The three triangular subdivisions are all based on projecting the sphere onto an octahedron, and then recursively subdividing the resulting eight triangles. There are many ways to do this; the four considered here are the methods proposed by Goodchild and Yang (1992), Dutton (1998), and Bartholdi and Goldsman (2000); the latter proposed both a quaternary and binary method of subdivision.

The natural sort works relatively well for the GPS stations because in many cases nearby antennas have similar site codes, and for the cores because nearby ones were often collected in sequence on a particular cruise. The sort by one or the other coordinate does better in all cases, and the various triangular subdivisions do even better, with the best results coming from the two proposed by Bartholdi and Goldsman (2000), perhaps not surprisingly since their methodology tries to label the codes in accordance with a space-filling curve. They found the binary division, which more closely approximates the Sierpinski space-filling curve, to produce fewer large jumps than the quaternary division; this is reflected in the better performance of the binary method for the cities dataset in Table 1. The GHAM code works as well or better than any of these methods; for all the actual datasets it consistently places nearest neighbors adjacent 60% of the time, and within 5 entries of each other 80% of the time.

4. Summary and conclusion

Kimerling et al. (1999) and Clarke (2002) have listed ideal properties of global gridding systems. The GHAM code satisfies some of them: it is global, arbitrarily precise, easy to describe and simple to compute, and compact in its final form. It does not satisfy some criteria that are needed for grids but not for addressing schemes (all cells the same shape), and some others that are desirable but unobtainable, such as use by a government authority. The property most stressed here, that of providing a good way of sorting large geographic datasets, is not mentioned in these articles, perhaps

because of their focus is on grids rather than addressing. Perhaps the most problematic attribute is that the codes, while easily read and compared, do not provide an intuitive link between code and location. This is however true of most global systems, even geographical coordinates. A recent effort to provide a combination of intuitive labels and precise position codes (Clarke et al., 2002) illustrates the problem: to use familiar labels requires a complex algorithm with many arbitrary settings to hold familiar place names. The simplicity and "sortability" of the GHAM code make it, not the perfect addressing system (it is not clear that there can be a perfect one), but one which is easy to implement and will be useful in examining large sets of geographic data.

Acknowledgements

I thank Michael Scharber for prompting me to (fruitfully) revisit the method of computing the GHAM code, and suggesting names for subroutines. The GPS archiving effort which gave rise to the GHAM code was funded by the US Geological Survey through the Southern California Earthquake Center (SCEC). This is SCEC contribution No. 865.

Appendix. A sample GHAM code

Table A1 illustrates the computation of the GHAM code for coordinates $\phi = 32.867772°$N, $\lambda = 117.252331°$W, for which the initial x and y are 0.17429908 and 0.38567552. The lines in Table A1 show the values of r_x, r_y, i_x, and i_y for each successive stage of multiplication by 16 and removal of the integer part. Each line then gives the code for the given i_x and i_y, from Fig. 2. The Level-6 code for this location is `E4I8U3W2V7I3`.

Table A1
GHAM code computation

Level	r_x	r_y	i_x	i_y	
1	2.78878529	6.17080836	2	6	E4
2	12.62056462	2.73293380	12	2	I8
3	9.92903396	11.72694087	9	11	U3
4	14.86454329	11.63105399	14	11	W2
5	13.83269262	10.09686379	13	10	V7
6	13.32308196	1.54982067	13	1	I3

References

Bartholdi, J., Goldsman, P., 2000. Continuous indexing of hierarchical subdivisions of the globe. International Journal of Geographic Information Science 15, 489–522.

Clarke, K.C., 2002. Criteria and measures for the comparison of global geocoding systems. In: Goodchild, M.F., Kimerling, A.J. (Eds.), Discrete Global Grids: A Web Book. University of California, Santa Barbara, http://www.ncgia.ucsb.edu/globalgrids-book.

Clarke, K.C., Dana, P.H., Hastings, J.T., 2002. A new world geographic reference system. Cartographic and Geographic Information Systems 29, 355–362.

Dutton, G.H., 1998. A Hierarchical Coordinate System for Geoprocessing and Cartography. Springer, Berlin, 230pp.

Engdahl, E.R., Villasenor, A., 2002. Global seismicity: 1900–1999. In: Lee, W.H.K., Kanamori, H., Jennings, P.C., Kisslinger, C. (Eds.), International Handbook of Earthquake and Engineering Seismology, Part A. Academic Press, Orlando, Florida, pp. 665–690.

Goodchild, M.F., Yang, S.R., 1992. A hierarchical spatial data structure for global geographic information systems. Graphics and Models for Image Processing 54, 31–44.

Kimerling, A.J., Sahr, K., White, D., Song, L., 1999. Comparing geometric properties of discrete global grids. Cartographic and Geographic Information Systems 26, 271–287.

Mark, D.M., 1990. Neighbor-based properties of some orderings of two-dimensional space. Geographical Analysis 22, 145–157.

Merrill, G.K., 1986. Map location literacy—how well does Johnny Geologist read? Bulletin of the Geological Society of America 97, 404–409.

Sahr, K., White, D., Kimerling, A.J., 2003. Geodesic discrete global grid systems. Cartographic and Geographic Information Systems 30, 121–135.

Tobler, W., Chen, Z.-T., 1986. A quadtree for global information storage. Geographical Analysis 18, 360–371.