Using Gray codes as Location Identifiers

Thomas Strang, Armin Dammann, Matthias Roeckl, Simon Plass
German Aerospace Center (DLR)
Institute of Communications and Navigation, Oberpfaffenhofen, Germany
firstname.lastname@dlr.de

ABSTRACT

In this paper we introduce a new location identification scheme based on the idea of Gray codes. The new scheme is particularly useful for applications which do have to cope with some level of inaccuracy or uncertainty. The applicability of the scheme is shown with some examples from the security domain, however is not restricted to this domain.

1. INTRODUCTION

The location is one of the most relevant properties of an entity in any Ubiquitous Computing environment. Having access to the location of a person or an object gives rise to a huge set of applications, while the availability of a location information is an opportunity and a challenge at the same time. Many location identification schemes (location taxonomies) have been introduced in the past to be used as representation format and unit of operation. Among them are geometric location identification schemes such as the WGS84 or UTM system, symbolic location identification schemes such as "Miami Airport, Terminal 4", as well as hybrid ones such as Leonhardt's Semi-Symbolic Hierarchical Location Model [1]. All of them have different advantages and disadvantages if used in the diverse algorithms adopted in all areas of applications in Ubiquitous Computing environments. However, due to its ease of use from the algorithmic side as well as global availability from the sensor side, namely GPS, the most dominant representation format is the WGS84 coordinate reference system.

For some applications a very accurate absolute location information is crucial, in particular for safety-of-life applications such as aircraft landing. For other applications accurate relative location information is required, for example to implement autonomous vehicle following ("platooning") [2], where the absolute location of the vehicles involved is less relevant than the relative distance between the vehicles. Yet another set of applications can also cope with some inaccuracy or uncertainty in absolute or relative location information, as long as some other criteria are met. A good example for the latter one might be the plausibility check of a location-aware authentication protocol.

This paper introduces a new location identification scheme particularly useful for applications which do have to cope with some level of inaccuracy or uncertainty. The new scheme can be perfectly mapped to and from other geometric reference schemes such as WGS84. The hierarchical coding provides means for both, the mapping to other reference schemes as well as to address the level of uncertainty of location information when used in an application.

2. APPLICATIONS

The general question any authentication protocol tries to answer is a derivative of *"Is the entity identifying itself as Alice really Alice?"*. Location-aware authentication schemes do make use of location identifiers as (part of) the dynamic properties of the entity (person, object) to authenticate – they try to answer a derivative of *"Assuming the entity identifying itself as Alice is Alice, can Alice currently be at the location reported for Alice, given some confirmed knowledge about the current or a previous location of Alice?"* as part of the authentication process. The plausibility check of the comparison between the current and a previous location is based on the assumption, that no person or object can travel a certain distance in less than a certain amount of time. All location-aware authentication schemes do have to cope with some level of inaccuracy or uncertainty, e.g. due to the limitations of the positioning technology, the time passed since the last confirmed location of Alice etc.

Such a location-aware authentication scheme has been introduced for instance with the CLARA protocol [3]. Here the enrichment was in its core a replacement of the key-independent message digest (MD) functions $H(m)$ used in standard RSA and DSS/DSA [4] signature protocols by a key-dependent message authentication code (MAC) of the form $H(m, k)$, e.g. HMAC as defined in RFC 2104, where a location identifier is used as the key (see figure 1 for an example). The location identifier itself can be taken from any arbitrary location taxonomy, for example a WGS84 coordinate such as "28.61N 80.61W".

The advantage of the approach is its ease of use and low computation complexity, however, a disadvantage is its susceptibility to inaccuracies or uncertainties in the location information. Even minimal differences between the location information used in the signature step compared to the one

Extended DSA-signature (r, s) of a message m with private key x, random number $k < q$ and a location identifier:
$r := (g^k \ mod \ p) \ mod \ q$
$s := (k^{-1}(H(m, \textbf{location}) + xr)) \ mod \ q$

Verification of the extended DSA-signature (r, s) with the location identifier:
$w := s^{-1} \ mod \ q$
$r =_? ((g^{(H(m,\textbf{location})*w) \ mod \ q} * y^{(rw) \ mod \ q}) \ mod \ p) \ mod \ q$

Figure 1: Geographic extended DSA signature procedure

in the verification step simply prevent the authentication scheme to work – the location information to be used in both steps has to be *identical*. But in real life applications, there are many reasons why the locations used to sign and verify might be geographically close but not identical. Reasons for this include GPS errors, different types of location sensors with different accuracy or simply movement of the entity to authenticate. So what is needed is a location-aware authentication scheme which is resistant to location inaccuracies and uncertainty up to a defined extent. The accepted level of divergency should not be fixed but a parameter of the location-aware authentication scheme.

This problem can be addressed using a hierarchical location coding scheme, which has similar characteristics on the bit-level like on the geography. The distance relationship of geographic locations should be preserved through the mapping of the new location code, meaning that two locations which are geographically close should be close in terms of location identifiers given a suitable metric. Such a scheme is introduced in the next section.

3. CODE

We propose to use a Gray code [5] to encode location identifiers. A Gray code is an ordering of 2^n binary numbers such that only one bit changes between neighboring code values. Of particular interest in this work is a specific Gray code and a recursive creation algorithm for this specific Gray code applying a technique known from quad trees [6], more precisely an alternating binary split approach. Starting with an 1-bit-code, the hierarchical 2D n-bit-code for any arbitrary value of n with a maximum Hamming distance[1] of 1 between any two direct horizontal or vertical neighbors in the 2D-grid can be created with the following binary alternating split algorithm:

1. Split an entire 2D area logically in two halfs and assign the value 0 to the left and 1 to the right half (cf. top-left part of Figure 2).
2. To create an n-bit-code from an $(n-1)$-bit-code, distinguish whether $(n-1)$ is odd (1,3,5,..) or even (0,2,4,..). If $(n-1)$ is odd, split every field horizontally, otherwise vertically. This results in a new separation for an n-bit-code with two times the number of fields of an $(n-1)$-bit-code (cf. top-right, bottom-left and bottom-right part of Figure 2).
3. With each split (horizontal or vertical), add the alternating pattern (0 and 1) or (1 and 0) in front of the binary value of the respective field of the $(n-1)$-bit-code (cf. also top-right, bottom-left and bottom-right part of Figure 2).

[1]The Hamming distance of two binary values is defined as the number of different bits.

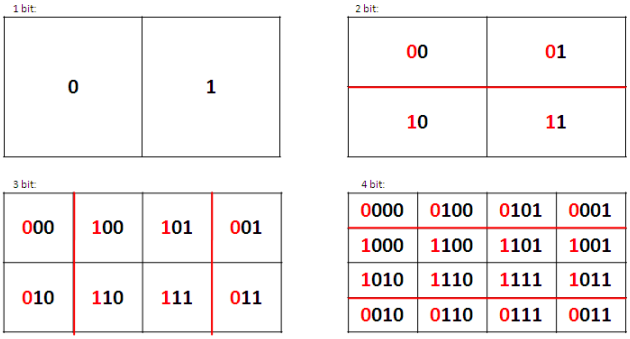

Figure 2: Gray code creation by alternating binary split

This 2D Gray code can be directly applied as location identification scheme to 2D area maps such as a Mercator projection of the earth's surface as illustrated in Figure 4. For example, the WGS84 Lat/Lon coordinate "28.61N 80.61W" would be mapped to the 22-bit-code value "10.0001.0000. 1100.1011.1100". The same WGS84 coordinate would be mapped to the 20-bit-code value "0001.0000.1100.1011.1100".

When calculating the distance of two code values in the sense of location identifiers, an appropriate metric (or distance function) has to be used. The Mannheim metric [7] preserves the desired characteristics of the Hamming distance as well as the cyclic continuation of the Mercator projection, i.e. that the most right location identifiers of the map have to be in minimum distance to the most left location identifiers for closing the sphere of the earth's shape. This is given by the cyclic definition of the Mannheim metric (in contrast to the Manhattan metric for example), i.e., a periodic continuation of the used code alphabet exists. Therefore, the Mannheim metric includes also the "way across the border" in 2D. Calculating the geographic distance using the Mannheim metric is for most applications a sufficiently suitable approximation of the "Loxodrome" (see Figure 3) most commonly used for navigation based on the Mercator projection. However, it should be noted that the Euclidean distance as used to determine the shortest distance on the earth's surface along a great circle ("Orthodrome", see Figure 3) varies from the Mannheim distance.

A Gray code created with the algorithm introduced in this section has the desired characteristics that the more geographically "apart" two arbitrary chosen code values are, the higher is the Hamming distance between the two code values in general. More precisely, the code guarantees that the hamming distance between two arbitrary chosen code values is always $\leq$ than the distance between the locations represented by the code values applying the Mannheim metric:

$$HammingDist(loc1, loc2) \leq MannheimDist(loc1, loc2) \quad (1)$$

Note that Gray codes of four or more bits are not unique: any code value has four neighbors with whom it joins an edge (taking wrap arounds into account). Using Gray codes of length $n > 4$ bits, for a given code value x, there are n other code values with Hamming distance 1 to x. All these n

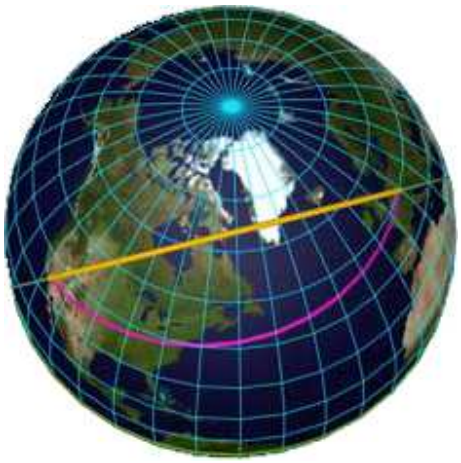

Figure 3: Orthodrome (yellow) and Loxodrome (purple) [Image: kowoma.de]

code values should be neighbors of x, but there can be only four. Thus there cannot be a permutation for $n > 4$ bits where the hamming distance between two arbitrary chosen code values is always equal to the distance between the locations represented by the code values applying the Mannheim metric. However, the code introduced here guarantees the $\leq$ relation as expressed in equation 1.

4. ANALYSIS

Location-aware applications which do have to be resistant to location inaccuracies and uncertainty can particularly make use of the hierarchical and distance preserving properties of the Gray code introduced in the previous section. Assume a situation as indicated in the applications above: the location identifiers used to sign (location_{sign}) and verify (location_{verify}) might be geographically close but not identical, for instance if location_{sign} was derived with a GPS, whereas location_{verify} was derived using the location of the cell tower of the user's GSM phone. Further, it is assumed that the difference between the two location identifiers is not already masked by the limited precision of too few digits after the decimal point of Lat/Lon values in the WGS84 format delivered by GPS. The first approach to add some tolerance to this location uncertainty is to go up 1 level in the code hierarchy. For Lat/Lon pairs this means to "sacrifice" 2x one digit after the comma in the decimal system. In the Gray code introduced in the last section going up one level means to ignore 2x the leftmost (least significant) bit. For any given n-bit-code, this is a well defined area with a fix height and a maximum width[2]. Table 1 gives an overview of the areas covered by a Gray code cell depending on the length of the code. Keep in mind that the difference between the Mercator projection and the real shape of the earth is smallest at the equator[3] and most significant near the poles[4].

[2]The real width of the area covered by one Gray code cell relates with $cos(lat)$ to the latitude and is only at the equator equal to the maximum.

[3]Circumference of the earth at the equator $= 2\pi r$ with mean earth radius $r_{equator} = 6.378{,}137$ km calculated to 40.075 km

[4]Mean distance from pole to pole along any longitude on the surface $= 2\pi r/2$ with mean earth radius $r_{poles,mean} = 6.371{,}001$ km calculated to 20.015 km

bits	split	max. width	height
	all area	40.075 km	20.015 km
1	2x1	20.037 km	20.015 km
2	2x2	20.037 km	10.005 km
3	4x2	10.018 km	10.005 km
4	4x4	10.018 km	5.003 km
5	8x4	5.009 km	5.003 km
6	8x8	5.009 km	2.501 km
7	16x8	2.504 km	2.501 km
8	16x16	2.504 km	1.250 km
...	...	...	...
15	256x128	156 km	156 km
16	256x256	156 km	78 km
...	...	...	...
23	4.096x2.048	9 km	9 km
24	4.096x4.096	9 km	4 km
...	...	...	...
29	32k x 16k	1 km	1 km
...	...	...	...
31	64k x 32k	610 m	610 m
32	64k x 64k	610 m	305 m
...	...	...	...
37	512k x 256k	76 m	76 m
...	...	...	...
47	16.7M x 8.3M	2 m	2 m
48	16.7M x 16.7M	2 m	1 m
...	...	...	...
odd n	$2^{(\frac{n+1}{2})}$x$2^{(\frac{n-1}{2})}$	$\frac{40.075}{2^{(\frac{n+1}{2})}}$km	$\frac{20.015}{2^{(\frac{n-1}{2})}}$km
even n	$2^{(\frac{n}{2})}$x$2^{(\frac{n}{2})}$	$\frac{40.075}{2^{(\frac{n}{2})}}$km	$\frac{20.015}{2^{(\frac{n}{2})}}$km

Table 1: Area covered by Gray code cell of different bit lengths

That fact makes the Mercator projection highly unsuitable to be used for most applications near the poles. It is thus not a problem that the Gray code introduced in this paper does not support routes or distances, which go precisely through the poles. Noteworthy is that each n-bit-code with odd n (i.e, $n = 1, 3, 5, \dots$) results in an area split where each code cell represents a square area, which is a more natural segmentation of area maps for many applications (and humans) than rectangular areas as with n-bit-codes with even n.

A second feature adding tolerance to uncertainty and the real advantage compared to the usual approach is – in addition to or instead to the cut of some digits/bits as just described – to use the core property of Gray codes, namely the minimum hamming distance between any two neighbors. So even if location_{verify} and location_{sign} are mapped into different Gray code cells, they are a maximum of 1 bit (direct neighbor in any of the 4 primary directions) respectively a maximum of 2 bits (corner neighbors) apart at the same bit-level. Given the same situation in the Lat/Lon encoding, the difference can be 1x one bit (best case) up to 2x any arbitrary number of bits different (worst case).

The following example illustrates these features for the example of the geographic extended DSA signature introduced in figure 1. Assuming the position of a signee can be determined using GPS with an average accuracy of 30 m. Encoding this position with a 37-bit Gray code constructed using the scheme above would result in a code value representing an area of 76 m along longitude and up to 76 m along latitude, covering the exact position of the signee and

an additional error of about 30-46 m in any direction. If the signature verification shall be used for plausibility check only, it might be even sufficient to check whether the signee has moved not more than, say, walking speed (ca. 3 km per hour) multiplied with the time since the last position update (e.g. 10 minutes), this would result in a further uncertainty of about 500 m in any direction. In this case a code value of 29 bit would be sufficient to be used as *location* key during signature and verification steps of the CLARA protocol. A significant advantage lies in the fact that the two location keys (37 bit, 29 bits) are identical in the last 29 bits, allowing for instance for a partial access scheme if not the exact 37 bit, but only the non exact plausible 29 bit code value can be used to successfully verify the signature.

5. RELATED WORK

Gray codes itself are not very new. Gray codes are for instance used as the code space of quadrature amplitude modulations (QAM) in wireless communication, which also allows for non-rectangular Gray code constellations.

Similarly, splitting of area/picture tiles into (usually four) subtiles using a quad tree approach [6] is not new. However, no work is known targeting the same design goals as with our binary alternating split algorithm, resulting in a hierarchical set of Gray codes of arbitrary length with the described features. Even the work of Mayrhofer and Spanring [8] who introduced a binary tile splitting algorithm along the same basic pattern for an entirely different purpose does not take care of proximity of the resulting code values nor on any specific metric. In particular, equation 1 does not hold for the code resulting from the algorithm of Mayrhofer and Spanring even if they would apply the Mannheim metric, which can be seen e.g. from the code values "00100" and "01110" on page 10 of [8].

The order of elements in our Gray code as constructed with our algorithm is similar, but not equal to Z-order as introduced by Morton [9]. Z-order represents a set of spacial indexes used in databases to support spacial queries. Z-order is known to be moderately locality-preserving, but not linear location preserving.

The work also somewhat close to ours is an approach to preserve location anonymity recently described by Horey at al. based on what they call "negative databases" [10]. In their approach the negative database consists of entries representing locations where an entity is *not* present. An evaluation of the approach as such is out of scope for this paper, but the code used to encode (non-)locations suffers from the problem that it is not distance preserving. This results in the effect that locations which are in fact very close are encoded with code values which are far more apart than others which represent locations which are not that close together.

6. SUMMARY AND CONCLUSION

In the previous sections we introduced a new location identification scheme based on the idea of Gray codes. The new scheme is constructed using a recursive algorithm, resulting in a hierarchical code where the hamming distance between two arbitrary chosen code values is always less or equal to the Mannheim distance between the tiles represented by the chosen code values. The main conclusion is that the code can be used to delimit the maximum inaccuracy or uncertainty as a certain maximum Hamming distance of two near but not identical location identifiers if the code is applied as a location identification scheme.

7. REFERENCES

[1] U. Leonhardt, "Supporting location-awareness in open distributed systems," Ph.D. dissertation, Department of Computing at the Imperial College of Science, Technology and Medicine, University of London, May 1998.

[2] M. Roeckl, K. Frank, T. Strang, M. Kranz, J. Gacnik, and J. Schomerus, "Hybrid fusion approach combining autonomous and cooperative detection and ranging methods for situation-aware driver assistance systems," in *19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2008)*, P. Chatzimisios and V. Kueh, Eds. IEEE, 09 2008. [Online]. Available: http://elib.dlr.de/54516

[3] T. Strang, "Geographische authentifikation und signatur (geographic authentication and signature)," in *3. Jahrestagung Fachbereich Sicherheit in der Gesellschaft für Informatik (Sicherheit 2006)*, J. Dittmann, Ed., vol. P-77, 02 2006, pp. 192 – 200. [Online]. Available: http://elib.dlr.de/22383

[4] B. W. Schneier, *Applied Cryptography*. Wiley, 1996.

[5] F. Gray, "Pulse code communication," u.S. Patent 2,632,058, filed 13 November 1947, issued 17 March 1953.

[6] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, pp. 1–9, 3 1974. [Online]. Available: http://www.springerlink.com/content/x7147683u3241843

[7] K. Huber, "Codes over gaussian integers," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 207–216, 1994.

[8] A. Mayrhofer and C. Spanring, "A uniform resource identifier for geographic locations ('geo' uri)," Internet Draft draft-mayrhofer-geo-uri-02, February 2008.

[9] G. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," IBM, Tech. Rep., 1966.

[10] J. Horey, M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks," *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2007)*, pp. 1–8, 08 2007.

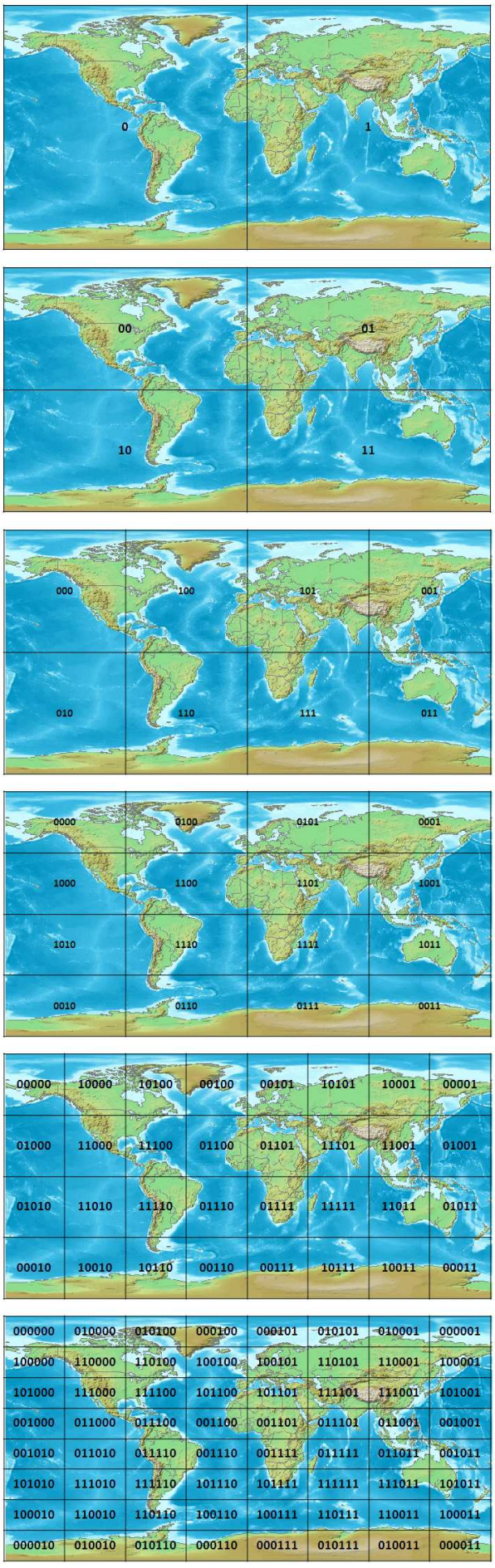

Figure 4: Gray-coded World Map