COMPARISON OF SPATIOTEMPORAL MAPPING TECHNIQUES FOR ENORMOUS ETL AND EXPLOITATION PATTERNS

R. Deiotte[1], R. La Valley[21]

[1]ISSAC Corp, 6760 Corporate Drive, Ste. 240, Colorado Springs, CO 80922, ray.deiotte@issaccorp.com
[2]OGSystems, Inc., 14291 Park Meadow Drive #100, Chantilly, VA 20151, Richard.LaValley@ogsystems.com

KEY WORDS: Spatiotemporal Encoding, Space-Filling Curves, Manifold Coverings, Computational Cost, Space-time Indexing, Space-time Encoding Efficiency, Space-time Encoding Utility

ABSTRACT:

The need to extract, transform, and exploit enormous volumes of spatiotemporal data has exploded with the rise of social media, advanced military sensors, wearables, automotive tracking, etc. However, current methods of spatiotemporal encoding and exploitation simultaneously limit the use of that information and increase computing complexity. Current spatiotemporal encoding methods from Niemeyer and Usher rely on a Z-order space filling curve, a relative of Peano's 1890 space filling curve, for spatial hashing and interleaving temporal hashes to generate a spatiotemporal encoding. However, there exist other space-filling curves, and that provide different manifold coverings that could promote better hashing techniques for spatial data and have the potential to map spatiotemporal data without interleaving. The concatenation of Niemeyer's and Usher's techniques provide a highly efficient space-time index. However, other methods have advantages and disadvantages regarding computational cost, efficiency, and utility. This paper explores the several methods using a range of sizes of data sets from 1K to 10M observations and provides a comparison of the methods.

1. MOTIVATION

With the recognition of Big Data problems over the past ten years, numerous technologies have emerged which provide cost effective and efficient data storage (e.g., Hadoop, MongoDB, HBase). These have been utilized to store huge volumes of data sourced from plethora systems, processes, and sensors. The volume of social media has exploded and, on any given day, Facebook is estimated to generate 6 billion new content items per day. Instagram is estimated to have over 2.4 Billion likes per day. Vine has over 1.44 Billion videos viewed per day, and Twitter users send around 500 million tweets per day (Carey, 2015). The volume of data has increased exponentially, and the velocity of data has increased dramatically as more new applications become available to the market. Overall, Mikal Khoso estimated in 2016 that 2.5 Exabytes of new data daily (Khoso 2016).

The true value of the increased spatiotemporal data occurs when the user can access and turn the data into actionable information. The sheer volume and availability of spatiotemporal data are increasing rapidly, and many methods of storage and analysis available for accessing and retrieving are not adequate for rapid recognition of the patterns of interest by the user. Current spatiotemporal mapping techniques provide unique capabilities for reduced storage size of complex data, rapid, intuitive comparative analysis, and novel pattern identification. Some issues confront the user regarding existing techniques in the form of computational cost, efficiency, and utility.

The goal of this paper is to extoll the virtues of spatiotemporal mapping techniques, expose their weaknesses and provide grounds to support the trades between cost, efficiency, and utility for the myriad use cases that leverage spatiotemporal data to gain insights and drive business decisions.

1.1 Motivational Metrics

We define the following metrics to aid in the comparison of spatiotemporal mapping techniques with the goal of covering the decision space for selection and employment of spatiotemporal mappings in diverse ETL and analytics ecosystems.

1.1.1 Computational Cost

When dealing with hundreds of millions or billions of records, all with some spatiotemporal data associated with them, we must concern ourselves with how much resources that are required to store, search, retrieve and compare these records. So when we reference cost in this paper, we refer to the amount of space it takes to store data, the time it takes to encode and decode information from one form to another and the complexity of performing proximity comparisons on the information.

(1) *We define the storage space measure of the cost metric as the number of bits required to store spatiotemporal data to a resolution of meters in space and seconds in time.*

---

[*] Corresponding author

(2) *We define the encode/decode measure of the cost metric as the number of seconds required to encode spatial or temporal data into a mapped/hashed form or to decode from the mapped/hashed form into standard spatial and temporal representations.*

(3) *We define the complexity of proximity measure of the cost metric as the number of mathematical operations required to assess the proximity in space and time of two events and locations.*

1.1.2 Efficiency

Though this paper, when we reference efficiency, we are assessing how well each method does the job it was intended to do. Measures like retained precision, propagation of error and uncertainty, introduction of edge cases, increased complexity in comparison, and preservation of relative locality all contribute to efficiency of spatiotemporal encoding methods.

(4) *We define the measure of precision of the efficiency metric as the required length of a mapping to achieve sub-meter and sub-second accuracy.*

(5) *We define the propagation of error measure of the efficiency metric as the amount of error introduced with each encode/decode operation cycle (from standard space/time representation to a mapped encoding and then back to space/time representation).*

(6) *We define the preservation of relative locality measure of the efficiency metric as the likelihood that two neighboring regions in space or time occur in neighboring regions in the encoding/mapping scheme.*

1.1.3 Utility

Because there is no point to changing the representation of data if that operation doesn't facilitate the delivery of information or the derivation of knowledge, we want to be able to assess how useful the mapping is for both human and machine methods, techniques and processes. When we discuss utility, we refer to measures like extensibility of the technique to other geographic domains, ease of human/machine interpretation, and the ability to support broad-spectrum analytics whether they are in the descriptive, predictive or prescriptive domains of analysis. While these measures are more subjective than those of Cost or Efficiency, they play a role in the calculus of selecting the right tool to support analytics objectives.

(7) *We define the measure of extensibility of the utility metric as the ability of the technique to be used for other geographic analytics domains. {none, low, medium, high; high values are the objective}.*

(8) *We define the ease of human/machine interpretation measure of the utility metric as the difficulty of interpreting the mapped information between two points or regions as opposed to leveraging the accepted representation of those points or regions {easy, moderate, difficult; easy values are the objective}.*

(9) *We define the ability to support analytics measure of the utility metric as the ability of the mapping to support multiple analysis activities without decoding,*

translating or transliterating the encoded/mapped information and without requiring augmenting information to support these analytics activities {low, medium, high; high values are the objective}.

2. BACKGROUND

2.1 Geohashing

One of the most widely used methods of geohashing (converting latitude and longitude into a single representative value) is that of Niemeyer (Niemeyer, 2012).

Essentially, Niemeyer's Geohash method encodes latitude and longitude as binary strings where each binary value derived from a decision as to where the point lies in a bisected region of latitude or longitudinal space. See Figure 1 for a graphical depiction of this.
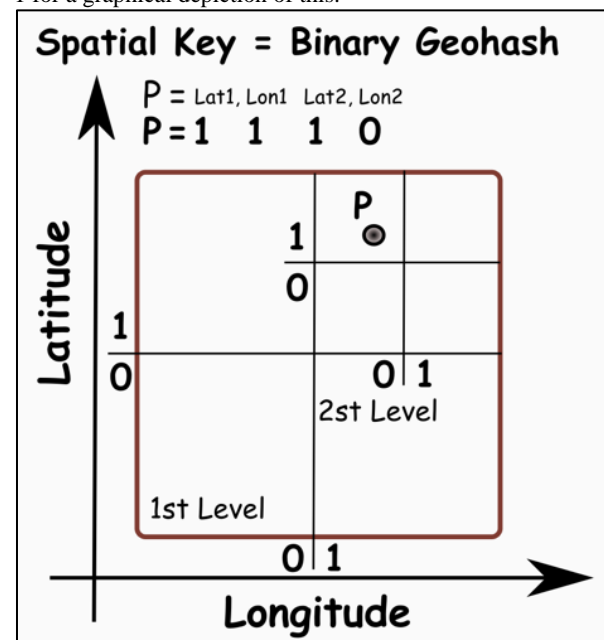

Figure 1. Binary Geohash

The encoded latitude and longitude binary string are interleaved (Figure 2), and the resultant binary string is encoded using a specialized 32-bit encoding schema.
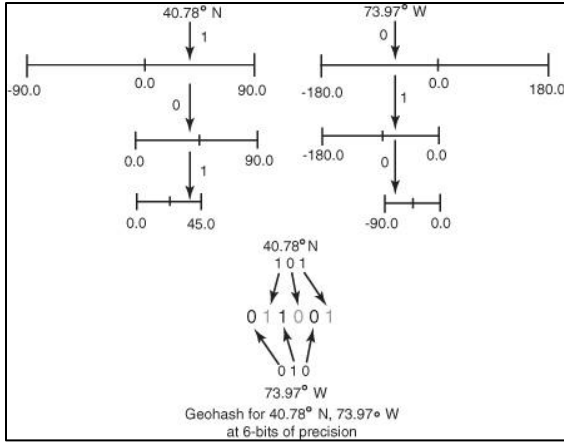
Figure 2. Niemeyer's Binary Interleaving for Geohash

The Niemeyer technique is similar to Morton encoding (Morton, 1966) (Figure 3) which is a specialized instantiation of a Z-order Space-filling curve (Figure 4) (Morton, 1966). Similarly, Natural Area Codes (NAC)(Shen, 2002) follow a similar encoding schema but employ a 30-bit encoding.

Figure 3. Morton Encoding

Niemeyer's technique has many useful features: rapid computation, a single-value string representation, variable precision through string truncation, proximal region detection, pattern support and easy human/machine interpretation.
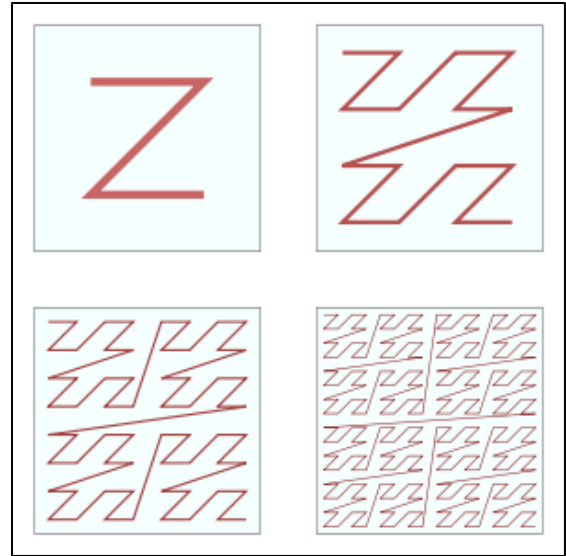

Figure 4. Four Iterations of Z-Order SFC

However, this technique has its limitations. The Niemeyer technique requires augmentative data to identify neighboring regions as the binary tree/Z-curve encoding is not regionally preserved (Figure 5), and neighboring areas can have wildly differing lead strings prompting users to question proximity without augmentative information), especially around the poles, the equator, and the prime meridian.

Additionally, the GeoHash encoding is lossy: every time values are encoded and decoded the accuracy of the data decreases. These capabilities and limitations will be explored empirically in Section 4.

Figure 5. Neighboring Region Incongruity

2.2   Timehashing

Timehashing or temporal encoding pioneered by Usher (Usher, 2010). Time hashing essentially follows a pattern similar to that of Geohashing with the noted differences that time is well-behaved (monotonically increasing, positive,

one-dimension, etc.). Usher's technique defines a span of 128 years (1970 to 2098) that is subsequently partitioned into eight equal bins, each mapped to a hexadecimal character (Figure 6).
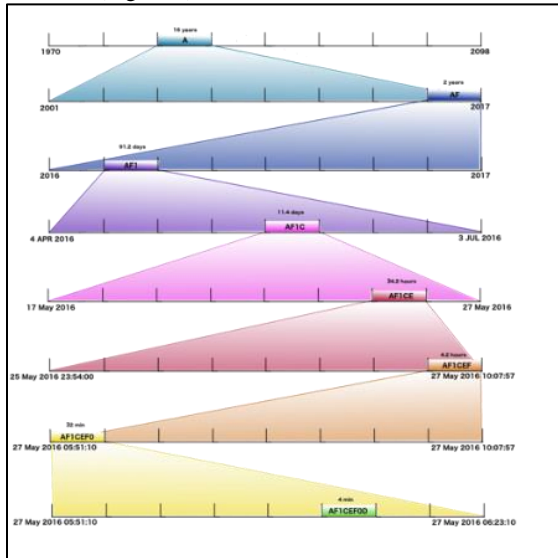

Figure 6. Timehashing Methodology

Usher's timehashing using sliding time windows allows users to define variable precision encodings of time and to compare them via string-matching algorithms. This method eliminates the need (as in Geohash) for complex comparison and similarity heuristics to be established ad hoc. This technique also allows for deep analytics support and simplified storage of complex values. Additionally, timehashing via Usher's technique maintains period proximity without border issues

There are drawbacks to this technique. The technique is lossy. The encoding paradigm is different than that of geohashing which precludes further spatiotemporal mappings (see next section). The boundaries of the mapping technique preclude hashing information before 1970 (for historical analysis) and beyond 2098 (fortunately we have 81 years to find something else!).

2.3 Spatiotemporal mapping

Unfortunately, there does not yet exist a mapping from Latitude/Longitude/Time space into a single, hashed value that shares the same traits as the hashes we have seen for space and time independently. However, by leveraging a set notation of hashes with both space and time, it is possible to "map" 3-space into 2-space and leverage the benefits of space and time hashing. The technique is currently being exploited in multiple communities to understand when two entities are staying together in one location, moving in space together, following one another, etc. The downside of this approach is, though, that we struggle with the same limitations as the original space and time hashes. Although this is an amalgamation of two techniques, we will be assessing it as that assessment potentially drives future research on the topic of single spatiotemporal hashes.

3. POTENTIAL ALTERNATIVES

We mentioned earlier that the typical Geohash algorithm is a relative of the Z-order space-filling curve. However, we may ask are other SFC useful to overcome the detractors of the current Geohash algorithm?" The answer is yes, but we have to make trade-offs as discussed in in sections 4 and 5. The Hilbert SFC is truly a mapping of n-dimensional space $[0,1]^n$ to a 1-dimensional line shown in Figure 7.
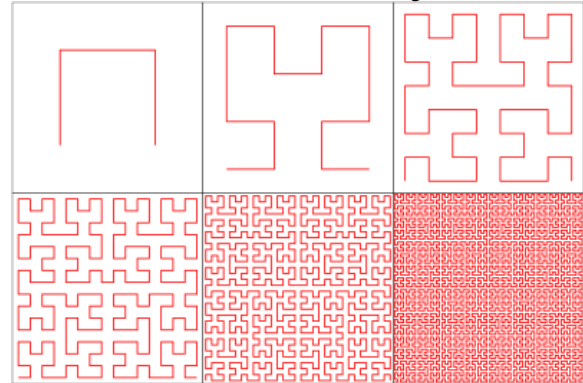

Figure 7. Six Iterations of the Hilbert SFC

As opposed to the Z-order curve, the quaternary location on the SFC can be encoded to a hash or a Gray-like encoding utilized. Because we are using the actual SFC to map 2-space locations to 1-space, we can benefit from the complexity, as well as their performance in measures defined above precludes their assessment here. Numerous articles and papers have been written on the Hilbert SFC, and it has been shown that the complexity and locality preservation are typically better than other curves (Mokbal & Aref, 2002)(Moon, et. al., 1996).

Additionally, the Hilbert curve does not have the same discontinuities at the equator and prime meridian that Geohash does.

For this effort, we leverage the Google S2 libraries to map Latitude/Longitude space onto the Hilbert SFC. The authors of S2 provide a novel way of looking at the Hilbert SFC mapping as seen in Figure 8. Google S2 Hilbert SFC Mapping
. By following this method, we can take advantage of the Hilbert features, and test it against the Geohash algorithm.
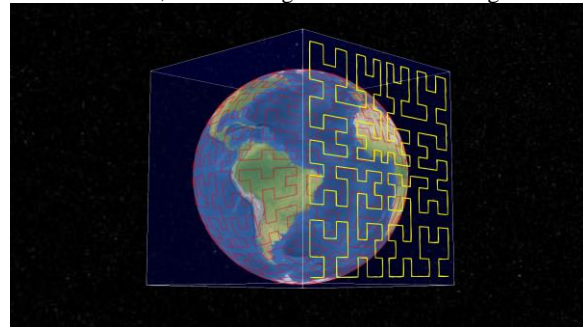

Figure 8. Google S2 Hilbert SFC Mapping

As one can imagine, the computational complexity of the Hilbert SFC goes up, but the trade-offs in precision, representation, accuracy and locality preservation may outweigh the cost of Hilbert SFC for many use cases.

4. ASSESSING VALUE

4.1   Computational Cost

The following sections illustrate our assessment and empirical support for those assessments against the raw Latitude/Longitude, Geohash and Hilbert SFC (S2).

4.1.1   Storage Cost Measure

Table 1 depicts the storage necessary between techniques.

Storage Cost					
Method	Type	Highest level of precision	Storage Space	1,000,000 Entry Space Requirement	Entry Space (MB
Raw Lat/Lon	Floating Point x 2	Sub-centemeter	128 bits (64x2)	128,000,000 bits	16MB
Geohash	String/Hash	Centemeter	72 bits	72,000,000 bits	9MB
Hilbert	String/Hash	Sub-centemeter	64bits	64,000,000 bits	8MB
Raw LLT	Floating Point x 3	Sub-centemeter, sub-second	192 bits	192,000,000 bits	24MB
Geohash w/ Time	String/Hash set	Centemeter/Subsecond	144 bits	144,000,000 bits	18MB
Hilbert w/ Time	String/Hash set	Sub-centemeter, sub-second	136 bits	136,000,000 bits	17MB

Table 1. Comparison of Encoding Storage Requirement

As can be seen, the Hilbert Encoding, even with the timehash, provides a smaller footprint than the raw data.

4.1.2   Encode/Decode Cost Measure

To demonstrate the speed-up/slow-down of these techniques, we generated random latitude and longitude points, globally, for 1,000, 5,000, 10,000, 50,000, 100,000, 500,000, 1,000,000, 10,000,000 locations. We then timed the execution for encoding and then decoded the positions for Geohash and Hilbert SFC. We performed this testing on an Apple MacBook Pro (2.2GHz Intel Core i7, 16GB Ram) and the results are shown in Figure 9.

As can be seen above, the Hilbert SFC ran about two orders of magnitude slower than the Geohash encoding mechanism. Depending on the use case, this delay may or may not be critical.
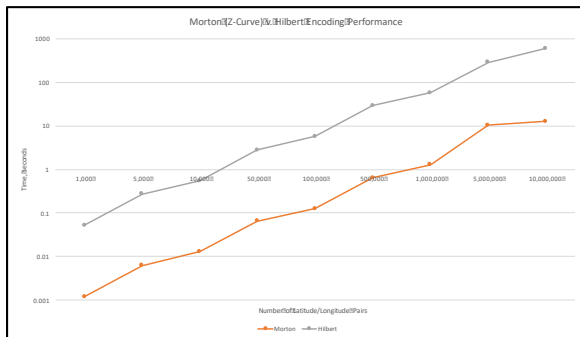

Figure 9. Comparison of Encoding Times for Geohash and Hilbert SFC

4.1.3   Complexity of Proximity

In determining the spatial or temporal proximity in any representation mechanism, there are operations that must occur to make that assessment. Some are more computationally complex than others. For example computing the proximity between two objects in Latitude/Longitude representation requires computing the Haversine (or other distance calculation) formula between the two objects and then determining if that distance is within some pre-established boundary. In Geohash space this comparison is made by comparing the first n characters of the string representation of position; this is a trivial string operation as opposed to the complex algebra of determining the even straight-line distance between two objects. A comparison of proximity complexity between methods is shown in Table 2. Complexity of Proximity Computations.

Complexity of Proximity			
Method	Number of Operations	Operations	Complexity Score
Raw Lat/Lon	3	Establish proximity boundary; Compute Distance; Assess Proximity	High
Geohash	2	Establish proximity boundary; Compare Appropriate Character	Low
Hilbert	2	Establish proximity boundary; Compare Appropriate Character	Low
Raw LLT	6	Establish spatial proximity boundary; Establish temporal proximity boundary; Compute spatial distance; Assess spatial proximity; Compute temporal distance; Assess temporal proximity	High
Geohash w/ Time	4	Establish spatial proximity boundary; Establish temporal proximity boundary; Compare Geohash character; Compare Timehash character	Low
Hilbert w/ Time	4	Establish spatial proximity boundary; Establish temporal proximity boundary; Compare Geohash character; Compare Timehash character	Low

Table 2. Complexity of Proximity Computations

4.2   Efficiency
4.2.1   Required Length of Hash

To determine the efficiency of each technique, we want to measure the length of the representation necessary to encapsulate the minimal error in the measurement of location. This assessment is shown in Table 3.

Required Hash Length			
Method	Length	Accuracy	Score
Raw Lat/Lon	128 bits	0 cm	0
Geohash	88 bits	0.5 cm	44
Hilbert	64 bits	0.1 cm	6.4

Table 3. Required Hash Length Comparison

Of note is the Raw Latitude and Longitude method: while scoring the best, one must consider that its introduced error is zero only because we are using floating point representations. So the size of the stored values are larger than other representations; one must also take into account the computations performed on the floating point representation of numbers. Rounding, machine precision, and other computer science issues impact these values in certain computations that the hashing methods are immune.

4.2.2   Error Introduction

As defined above, we need to be able to measure the amount of error introduced in encoding and decoding the hashed representations. Because the encoding is not precise, we inherently add some uncertainty to the measurements every

time we encode them. If the data is encoded and decoded multiple times without retaining the original encoding, there is a potential for introducing compounding errors that are unrecoverable. This method can lead to a lack of confidence in data and computations and introduce risk into decision-making operations.

The assessment of Geohash and Hilbert encodings shown in Table 4.The third column refers to the number of encoding/decoding cycles have before there is a loss of two orders of magnitude in accuracy (e.g., how many encode/decode cycles does it take to move from centimeter resolution to meter resolution due to the error introduced in each encode/decode operation).

Introduced Error at Maximum Resolution		
Method	Error Introduced	Number of Cycles
Geohash	2 cm	50
Hilbert	0.1 cm	1000

Table 4. Error Rate Introduction Comparison

4.2.3 Preservation of Locality

Rather than reproducing the work of Mokbel and Aref (Mokbel & Aref, 2002), we will simply reference their work and make the statement that since Geohash follows a Z-order SFC, it performs worse than a Hilbert SFC encoding. That is to say; the Hilbert SFC encoding method preserves locality better than Geohash which makes comparison and neighborhood assessments easier and more accurate.

4.3 Utility
4.3.1 Extensibility

We believe that a spatiotemporal encoding scheme should be ubiquitous and able to handle locations above and below the surface of the earth. Because of the nature of Hilbert SFC's we can extend the representation of sub- and above-surface locations with relative ease. Unfortunately, if we want to use Geohash for anything other than Latitude and Longitude mapping, we are out of luck. The assessment is shown in Table 5.

Extensibility	
Method	Extensibility
Raw Lat/Lon	Low
Geohash	None
Hilbert	High

Table 5. Extensibility Comparison

4.3.2 Interpretation

To maximize utility the information being used by the human analyst or by the machines they employ, one must leverage an encoding that is easy to interpret. Hashing techniques have many advantages over raw latitude and longitude representations: single value representation, multiple location comparison, relative localities, variable, precision, and monotonic behaviors. However, getting familiar with hash representations takes some getting used to, and for the analyst hashing patterns can be learned and

snap assessments can be made just as they are in Latitude and Longitude space. Our assessment of the ease of interpretation is shown in Table 6. In using all three representations, they all have benefits and detractors, and for this reason, we believe that they are all similar on the scale of interpretation.

Interpretation	
Method	Interpretation
Raw Lat/Lon	Moderate
Geohash	Moderate
Hilbert	Moderate

Table 6. Ease of Interpretation Comparison

4.3.3 Analytics Support

For the measure of analytics support, we assess how well each encoding mechanism supports descriptive, predictive and prescriptive analytics and, to the extent possible, if the technique can be used without amplification information and excessive encode/decode cycles. Table 7 describes our assessment of each approach.

Analytics Support						
Method	Descriptive		Predictive		Prescriptive		Encode/Decode Cycles
	Support	Complexity	Support	Complexity	Support	Complexity	
Raw Lat/Lon	High	High	Moderate	High	Moderate	High	None
Geohash	High	Moderate	High	Moderate	Moderate	High	Some
Hilbert	High	Low	High	Low	High	Low	Few

Table 7. Analytics Support Comparison

4.4 Summary of Findings

Looking now at the metrics of computational cost, efficiency, and utility we see that the Geohash and Hilbert SFC are pretty competitive and the choice between the two, in our opinions, really boils down to how the user intends on leveraging the hashing method. This will be discussed briefly in the next section. Table 8 represents the metric findings for each technique.

Metric Comparison			
Method	Cost	Efficiency	Utility
Raw Lat/Lon	High	Moderate	Low
Geohash	Moderate	Moderate	Moderate
Hilbert	Moderate	High	High

Table 8. Metric comparison

5. TRADE-OFFS

In choosing an encoding scheme (or even if an encoding is needed), one must ascertain what the data is to be used for and how encoding (or not) will benefit their intended use. For fast running encoding that has few encode/decode cycles and computational overhead of proximity determinations is not a concern, the traditional Geohash is the method of choice. However, if you are dealing with large volumes of data that must be compared, patterned, encoded and decoded repeatedly, must incorporate sub- and above ground positions, and must support predictive and prescriptive behavioral and performance assessments, the Hilbert

encoding is the method to choose. For both Geohash and Hilbert encodings, some of the computational overhead can be overcome via parallelization and introduction of additional hardware. In the end, it is up to the analyst to determine which method to use, based on their needs and limitations.

6. CONCLUSIONS

We have discussed the limitations of current geo- and time-hashing algorithms established a set of comparative metrics for assessing encoding algorithms and demonstrated a viable comparison of techniques for spatiotemporal encoding. We find that, in general, the Hilbert encoding is preferred due to its post-encoding utility and computational cost. However, each analytics problem is different, and there are cases in which geohashing or leveraging raw latitude and longitude values are preferable to Hilbert encoding.

7. FUTURE RESEARCH

Based on the findings presented here and the limitations of current techniques we believe that there is a great deal of research yet to be done in unifying space and time and determining a ubiquitous method for encoding space-time.

REFERENCES

Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B. (1990). The R*-tree: an efficient and robust access method for points and rectangles (Vol. 19): ACM.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509-517.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.

Carey, George, How Much Data is Generated Every Minute on Social Media? WERSM, 2015

Finkel, R. A., & Bentley, J. L. (1974). Quad trees a data structure for retrieval on composite keys. Acta Informatica, 4(1), 1-9

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching (Vol. 14): ACM.
Kamel, I., & Faloutsos, C. (1993). Hilbert R-tree: An improved R-tree using fractals.

Khoso, M., How Much Data is Produced Every Day? Analytic Trends, Northeastern University Level Blog, March 13, 2016.

Malensek, M., Lee Pallickara, S., & Pallickara, S., (2013). Exploiting geospatial and chronological characteristics in data streams to enable efficient storage and retrievals. Future Generation Computer Systems, 29(4), 1049-1061.

Mokbel, M. F., Aref, W. G., & Kamel I. (2002). Performance of Multi-Dimensional Space-Filling Curves. GIS'02. McLean: ACM

Moon, B., Jagadish, H. V., Faloutsos, C. & Saltz, J. (1996) Analysis of the clustering properties of Hilbert Space-filling Curve. http://www.cs.umd.edu/TR/UMCP-CSD:CS-TR-3611

Morton, G. M. (1966). A computer oriented geodetic data base and a new technique in file sequencing: International Business Machines Company.

Niemeyer, G. Geohash-Wikipedia., 2012. https://en.wikipedia.org/wiki/Geohash (accessed May 10, 2017).

Rew, R., & Davis, G. (1990). NetCDF: an interface for scientific data access. Computer Graphics and Applications, IEEE, 10(4), 76-82.

Shen, X. The Official Web Site of the Natural Area Coding System. 2002. http://www.nacgeo.com/nacsite/ (accessed 5 10, 2017).

Tobler W., (1970) "A computer movie simulating urban growth in the Detroit region". Economic Geography, 46(2): 234-240

Usher, A. (2010). Temporal Algorithms for processing and analyzing large datasets. Sterling Data LLC Report.