Detecting Patterns using Geo-temporal Analysis Techniques in Big Data

Rich La Valley
Leidos
Arlington, Virginia

Abe Usher
HumanGeo Group LLC
Arlington, VA

Don Henderson
Henderson Consulting
Olney, Maryland

Paul Dorfmann
ISAS Consultant
Jacksonville, FL

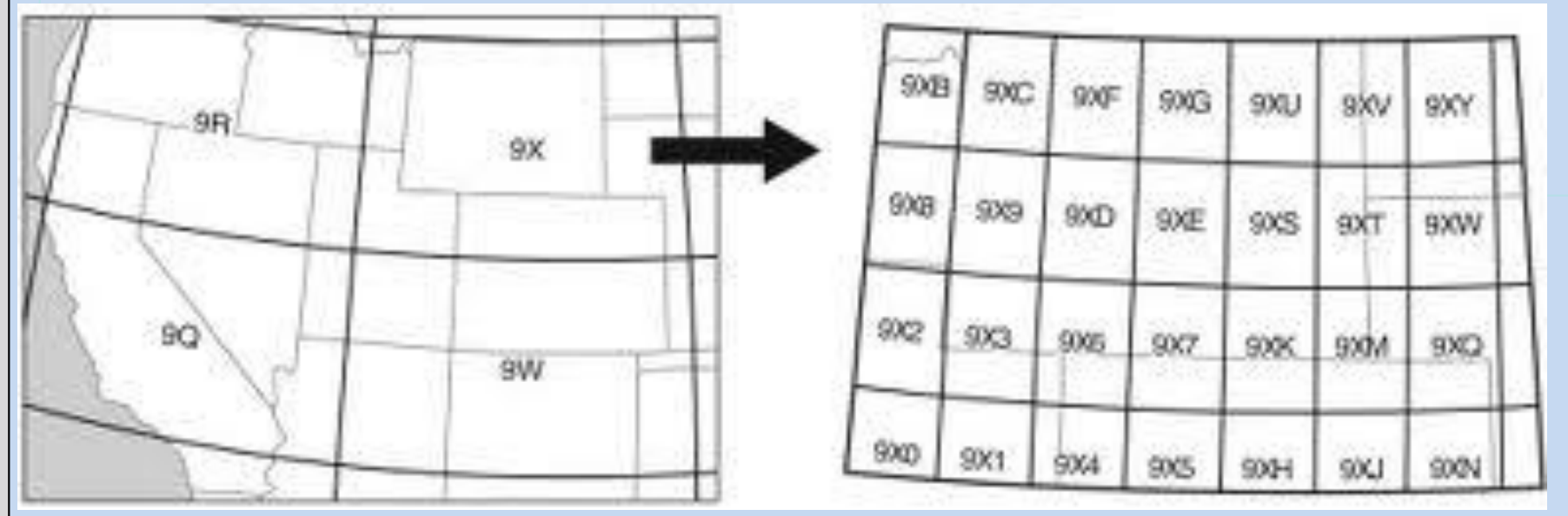
Abstract:

New innovative analytical techniques are necessary to extract patterns in Big Data which have temporal and geo-spatial attributes. An approach to this problem is required when geo-spatial time series datasets which have billions of rows and the precision of exact latitude and longitude data makes it extremely difficult to locate patterns of interest The usual temporal bins of years, months, days, hours and minutes often do not allow the analyst control of precision necessary to find patterns of interest. Geohashing is a string representation of two dimensional geometric coordinates. Time hashing is a similar representation which maps time to preserve all temporal aspects of date and time of the data into a one dimensional set of data points. Geohashing and time hashing are both forms of a Z-order curve which maps multidimensional data into single dimensions while preserving locality of the data points. This paper explores the use of a multi-dimensional Z-order curve combining both geohashing and time hashing that is known as "geo-temporal" hashing or "space-time" boxes using SAS®. This technique provides a foundation for reducing the data into bins that can yield new methods for pattern discovery and detection in Big Data.

GeoHashing:

Geohash is a latitude/longitude geocode system is a hierarchical spatial data structure which subdivides space into buckets of grid shape whicho offers properties like arbitrary precision and the possibility of gradually removing characters from the end of the code to reduce its size (and gradually lose precision).

Example



Latitude: 12.9637069 and **Longitude:** 77.6377931
4th Main Rd, Domlur II Stage, Bengalunu, Karnataka
tdr1wxyp5dn7v

Precision is simply eliminating values at the end of the string

tdr1wxyp5dn7v => (12.9637, 77.6378) – Domlur, Bengaluru, Kamataka
tdr1wxyp5dn7v => (12.96, 77.64) – Bengaluru, Kamataka
tdr1wxyp5dn7v => (13, 78) – malur-Bangarapettu

Why is this important? Throwing away precision (in a tunable fashion) allows for a much simpler approach to BIG DATA analysis at the scale of billions of events/observations

Timehasing

Timehash algorithm is a way of representing a user-defined variable precision temporal. It is an implementation of a quadtree that is defined with simple ASCII characters.

Example

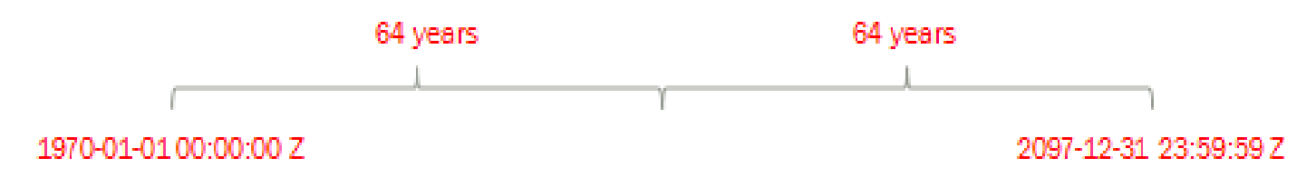
.

Interval: Create a fixed interval of time (e.g. 128 yrs, 1970-01-01 00:00:00 GMT to 2097-12-31 23:59:59)

Conversion. Convert the times to secons since epoch (1907-01-01) in Zulu time.

Subdivision. Subdivide the interval of seconds into a binary tree of intervals, represented by bits of time (e.g. +/- 128 year bit, +/ 64 year bit, +/- 32 year bit). Apply a sufficient number of bits to get an interval length sufficient for capturing the spans of time you want to represent (e.g. +/ 5 days, +/- 2 days)

ASCII Representation. Use a base 32 codemap of eight ASCII characters to represent the number of bits to encode the time interval

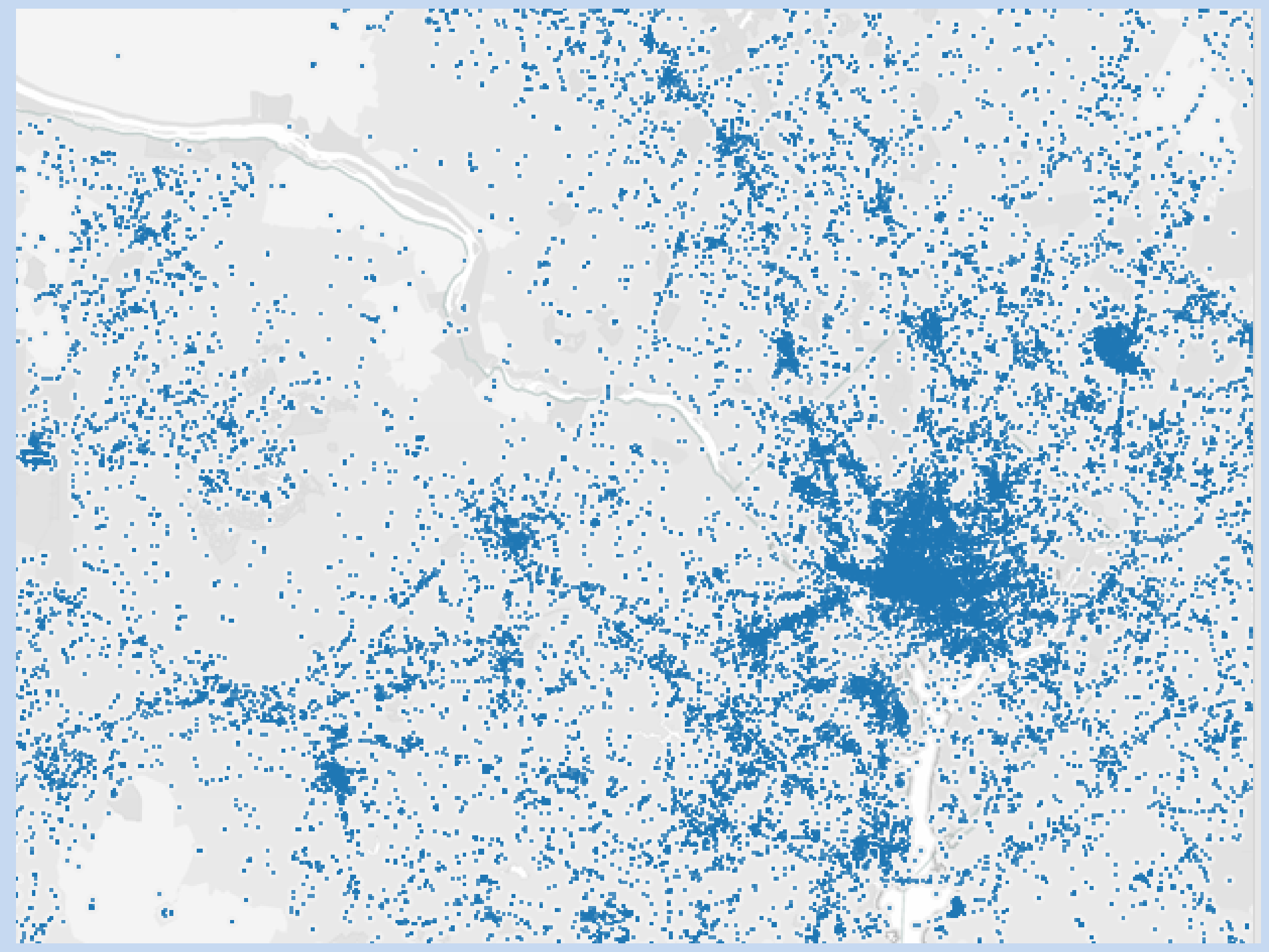


Date Time Group format: 2013-04023 10:53:59

- Add1efc1bd represents this timestamp within +/- 1.88 seconds
- Add1efc1b represents this timestamp within +/- 15 seconds
- Add1efc1 represents this timestamp within +/- 2 minutes
- Add1efc represents this timestamp within +/- 16 minutes
- Add1ef represents this timestamp within +/- 2.14 minutes

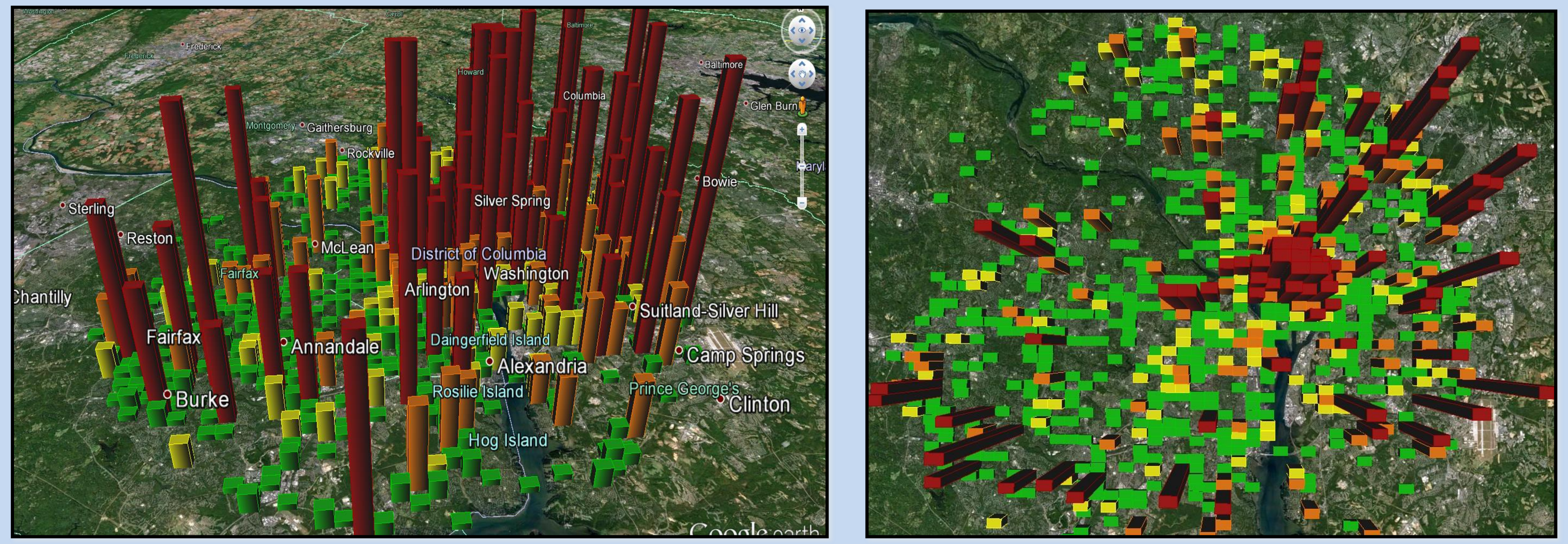
Why is this important? Throwing away precision (in a tunable fashion) allows for a much simpler approach to BIG DATA analysis at the scale of billions of events/observations

Geospatial Analysis of Twitter in DC area



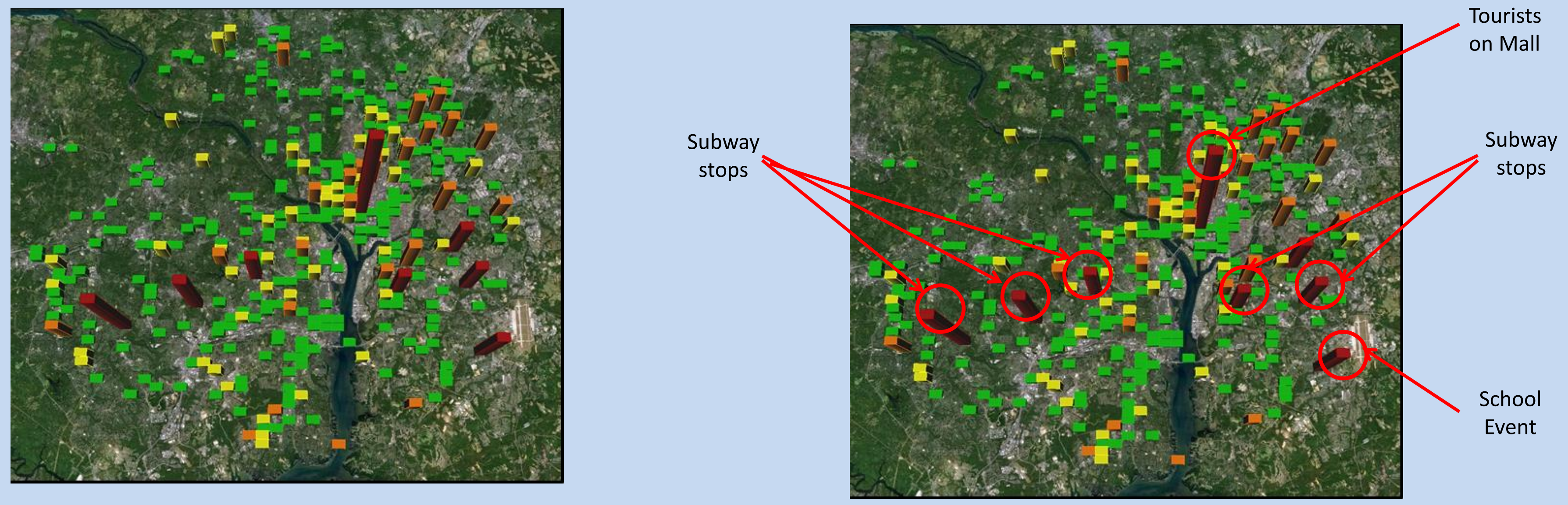
This is a typical geospatial visualization that is then used for Analysis. Patterns are somewhat visible with these 134K rows of data. The precision of most latitude and longitude data can sometimes mislead the analyst without knowing it. If the data is accumulated over several dates or months, then complex behaviors of a person or group of persons is often difficult to process. Specialized techniques and processing have been developed to handle these conditions but when there is more than 10 Million rows, the computations increase exponentially. Finding complex behaviors become nearly impossible.

Geohashing Analysis



Using Geohashing, the patterns become easier to see but behaviors which involve time can still be difficult to see.

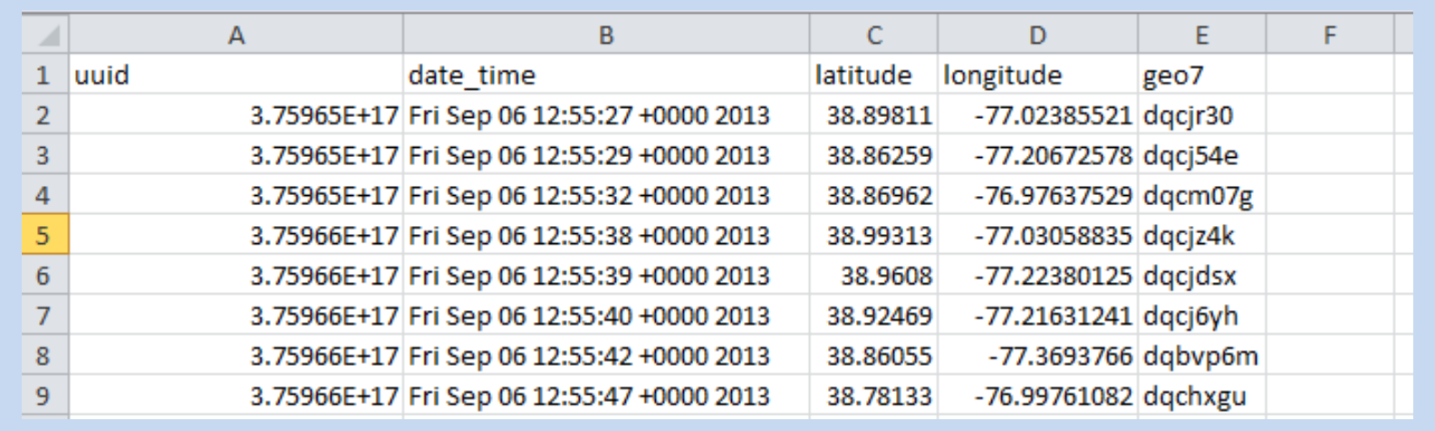
Geotemporal hashing example



Using temporal hashes, the patterns become easier to see and when your behavior of interest requires you to define specific periodicity (every 10 minutes), then the technique described is necessary when there is Big Data.

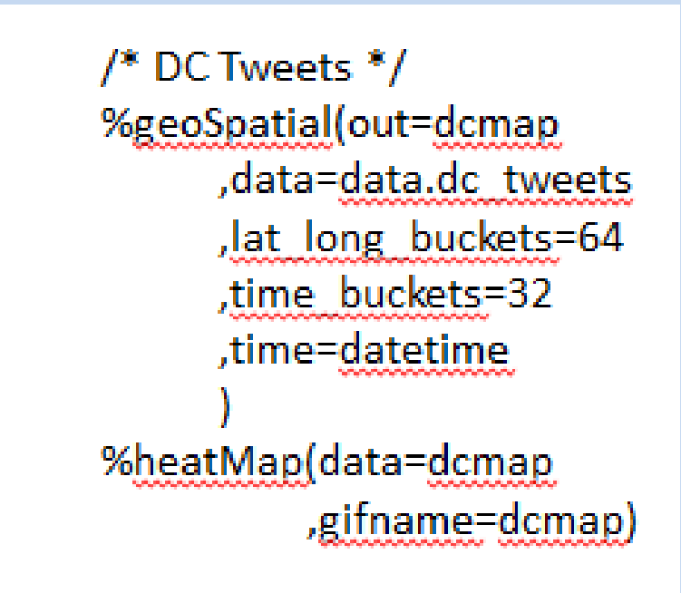
Geotemporal hashing with SAS®

Data Example



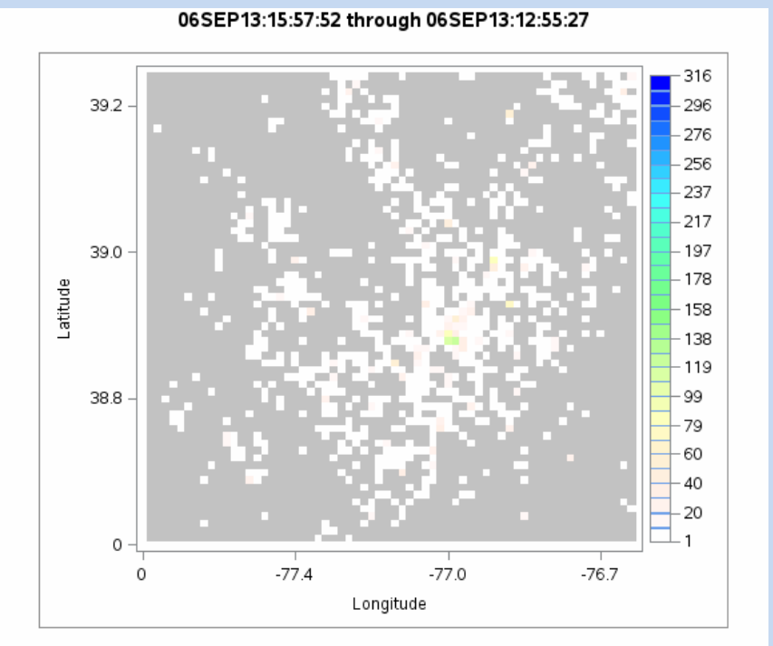
	A	B	C	D	E	F
1	uuid	date_time	latitude	longitude	geo7	
2	3.75965E+17	Fri Sep 06 12:55:27 +0000 2013	38.89811	-77.02385521	dqcjr30	
3	3.75965E+17	Fri Sep 06 12:55:29 +0000 2013	38.86259	-77.20672578	dqcj54e	
4	3.75965E+17	Fri Sep 06 12:55:32 +0000 2013	38.86962	-76.97637529	dqcm07g	
5	3.75966E+17	Fri Sep 06 12:55:38 +0000 2013	38.99313	-77.03058835	dqcjz4k	
6	3.75966E+17	Fri Sep 06 12:55:39 +0000 2013	38.9608	-77.22380125	dqcjdsx	
7	3.75966E+17	Fri Sep 06 12:55:40 +0000 2013	38.92469	-77.21631241	dqcj6yh	
8	3.75966E+17	Fri Sep 06 12:55:42 +0000 2013	38.86055	-77.3693766	dqbvp6m	
9	3.75966E+17	Fri Sep 06 12:55:47 +0000 2013	38.78133	-76.99761082	dqchxgu	

SAS Code



```
/* DC Tweets */
%geoSpatial(out=dcmap
        ,data=data.dc_tweets
        ,lat_long_buckets=64
        ,time_buckets=32
        ,time=datetime
        )
%heatMap(data=dcmap
              ,gifname=dcmap)
```

Geotemporal Hashing w/SAS



A literature review of the SUGI and SAS Global Forum proceedings did not find this technique and code was developed to be provided to the SAS community. The above example illustrates the type of data that is typicall y used, the code that was developed by several of the authors, and a simple animation of the results for geotemporal hashing for the DC Twitter data. This animation can be superimposed onto a similar map of DC and you see the types of patterns that you are interested. The full code will be posted on Don Henderson's page on sasCommunity.org.

Why is it Important?

This ePoster attempts to introduce the use of geotemporal hashing techniques to find patterns of life using data such as twitter. We showed some examples of this using Washington, DC area and we showed the SAS code that could be used to implement such a technique. Typically, geospatial visualization is used to look for patterns such as heat maps or just plotting of geospatial data on a map. Complex patterns become harder to discover when the data is collected over days, months, or years and then displayed and the computational requirements of finding these patterns is exponential as the number of observations grows. Beyond 10 million observations, new methods are required to facilitate the discovery of these patterns. The technique illustrated in this ePoster has been successfully used with applications with over billions of rows with the precision being defined by the user.

References

[1] Unwin, D. J. 1996 GIS, spatial analysis and spatial statistics', Progress in Human Geography 20(4): 540-441

[2] Z. Sheppard. Flickr blog: 5,000,000,000. http://blog.flickr.net/en/2010/09/19/5000000000/,2010.

[3} Box G E P, Jenkins GM, Reinsel G C 1994 *Time series analysis: Forecasting and control* (Singapore: Pearson Education Inc.)

[4] Chatfield C 1996 *The analysis of time series* 5th edn (New York, NY: Chapman and Hall)

[5] Juang B H, Rabiner L 1993 *Fundamentals of speech recognition*. (Englewood Cliffs, NJ: Prentice Hall)

[6] O'Shaughnessy D 2003 *Speech communications: Human and machine* (Piscataway, NJ: IEEE Press)

[7] Hauskrecht, M., Valko, M., Batal, I, Clearmont, G., Visweswaram, S., and Cooper, G., 2010. Conditional outlier detection for clinical alerting. In Proceedings of the American Medical Informatics Association (AMIA).

[8]SACCHI, L., LARIZZA, C., COMBI, C., AND BELLAZZI, R. 2007. Data mining with Temporal Abstractions: learning rules from time series. Data Mining and Knowledge Discovery.

[9]HO, T. B., NGUYEN, T. D., KAWASAKI, S., LE, S. Q., NGUYEN, D. D., YOKOI, H., AND TAKABAYASHI, K. 2003. Mining hepatitis data with temporal abstraction. In Proceedings of the international conference on Knowledge Discovery and Data mining (SIGKDD).

[10] JAIN, A., CHANG, E. Y., AND WANG, Y.-F. 2004. Adaptive stream resource management using Kalman filters. In Proceedings of the international conference on Management of data (SIGMOD).

[11] PAPAPETROU, P., KOLLIOS, G., AND SCLAROFF, S. 2005. Discovering frequent arrangements of temporal intervals. In Proceedings of the International Conference on Data Mining (ICDM).