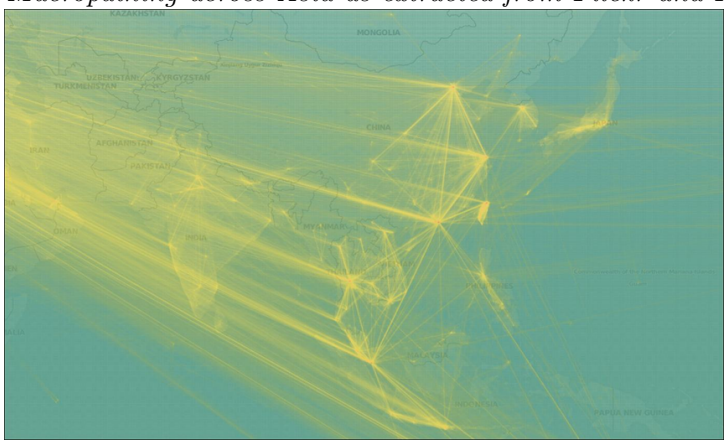
Movement Inference through Aggregate Trajectory Extraction

Jonathan Larson, Justin Gawrilow, Eric Kimbrel, Carla Schoepfle

Sotera Defense Solutions - 2012

Aggregate Macropathing across Asia as extracted from Flickr and Panoramio



Abstract

Geospatial and temporal data are becoming ever more ubiquitous as digital devices permeate society and better record everyday life. These sources of "Big Data" can be repurposed to expose aggregate patterns of activity. In this paper, we propose a novel distributed computing method for movement path extraction and aggregation over large data collections that contain both geospatial and temporal components. The results are benchmarked on small cluster and then several visualizations methods are proposed to enable deeper analysis.

1 Introduction

Movement data is everywhere around us and being captured by more devices as we push deeper into the digital age. Phones, GPS enabled wristwatches, cars, and other devices capture our progress in traveling between locations. Often times the data isn't even intended to capture movement, as in the case of online photo sharing sites. Someone takes a photo with a GPS enabled camera and takes another photo at a new location, creating a movement track of their activity. This sort of activity is a form of honest signal[1] that is captured during the normal operation of a geo-positionally aware device. In this paper, we propose a method termed Aggregate Micropathing or Aggregate Macropathing, which allows for massive scale distributed aggregation of movement patterns from data which contains geospatial, temporal, and device ID information.

Geo-temporal data is increasingly becoming more available and prolific. This has especially boomed with the advent of social networking and geo-aware cell phone devices. Services like Twitter[4], Flickr[2], Panoramio[3], Automatic Identification System (AIS)[7], OpenStreetMaps[5], and Foursquare[6] are among just a few of the services that record a device event in both time and space. Popular infographics experts, such as Eric Fisher, have performed aggregate analysis across many of these data sources as seen in [12]. However, this data can not only show baseline patterns of activity, but can also be used to characterize aggregate movement that occurs within the data.

Additionally, due to physical world constraints, we can discern and classify types of movement, such as vehicular, pedestrian, airplane, or a probably erroneous data point. For example, let us suppose Person A took a photo at Location Y. Five seconds later, Person A took another photo at Location Z. The distance between Location Y and Z is over 800KM. The laws of physics would highly suggest that this particular data was erroneous. Thus, improbable data can be filtered out using basic real world constraints.

It is our belief that by looking at the activity of all these devices in aggregate, we can infer honest signals from the data to learn more about the movement patterns within a given region. We have many individual and small samples of movement in a region. Within a given city, a couple photos taken in sequence from a pair of tourists, a local real estate agent documenting houses for sale, or someone posting to their twitter account between business meetings. Each of these events provides a movement trajectory that in of themselves isn't very

interesting. However, when combined together in aggregate, looking at these very small movements, especially those small movements of only a few feet over the course of a few seconds, can provide valuable observations. As seen in prior research, movement patterns between neighborhoods[8] and cultural boundaries[9] can be detected. Additionally, we have found that taking a global view on this data using the same techniques can also provide powerful contextual information. These wide area views can expose travel pattern behaviors between major hubs and cities to show a macro level of traffic flow, purely based on public open source data that was not originally built for this purpose.

Aggregate Micropathing is a technique that aims to unlock some of these honest signals and go beyond simply plotting the data. Instead, this technique looks to derive and infer patterns of movement from the data. This technique also provides benefits in that it allows for the inference of movement in areas without data coverage.

1.1 Related Work

We envision that the proposed appraoch would fit well into existing trajectory visualization research and interactive navigation that is presented in [**?**]. Trajectory based patterns and their extraction in aggregate from GPS-equipped devices is also discussed in the compelling examples presented in [15]. However, this focuses on movement patterns between regions of interest. An approach for an assortment of event analysis, which uses clustering, spatial, and temporal analysis using Flickr and Panoramio is shown in [13]. Trajectory clustering and visualization is presented in /citeRinzivillo as this paper also researches aggregate movement patterns. Travel sequences are discussed in [17].

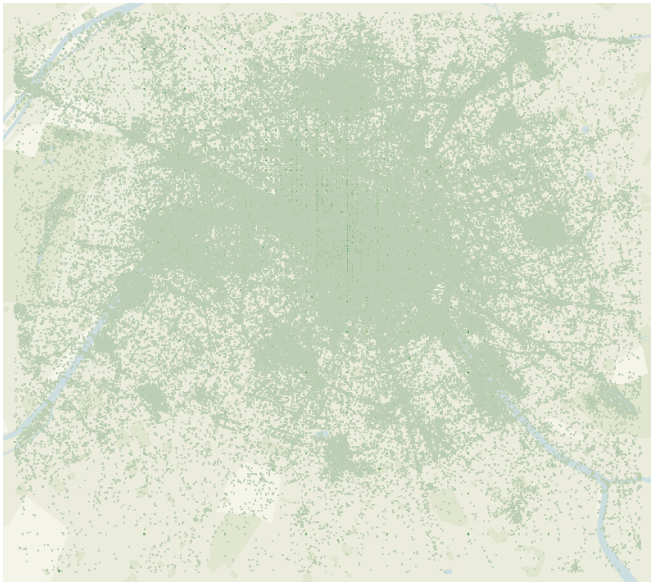


Figure 1: Flickr Data in Paris without Binning

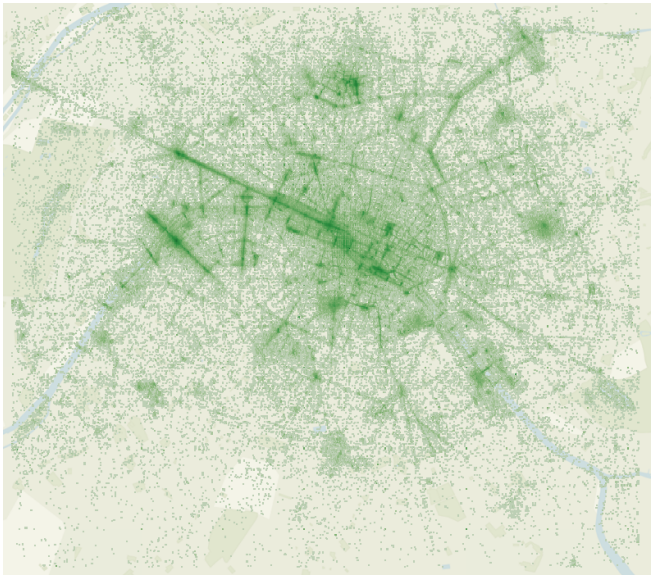


Figure 2: Flickr Data in Paris with Binning every .0002 latitude and .0002 longitude

2 Implementation

This section describes the structure and process underlying the Aggregate Micropathing technique that allows for movement extraction and representation.

Before providing a detailed explanation on the technique, let us first review the common binning technique of aggregated heatmaps. Consider the two following snapshots that visualize around 2 million pieces of photo metadata from Flickr in Paris. Figure 1 shows the raw data just plotted on a map with each photo plotted at its exact coordinate. The increased volume of photos at a coordinate is represented by a darker green color. However, due to the high precision of each coordinate, the likelihood that two photos were taken in precisely the same location is unlikely. Therefore, we have a painting problem if multiple photos are in almost the same place as the pixels will overlap one another instead of reflecting an increased intensity in the general region. Now consider the same data with nearby events binned at a carefully chosen resolution in Figure 2. This shows a much clearer picture of Paris and illuminates major roadways and thoroughfares. This requires some expertise in the selection of the bin size as selecting too fine or too coarse a bin size may obscure important features within the data.

However, while these simple binning strategies may provide situational awareness, they often lack information on changes over time and almost always lack information on movement patterns. To extract these details from the data, a four stage process is taken to process this data further.

The raw data should preferably have at least four key fields: latitude, longitude, datetime, and an ID. The ID field is typically a device id or a user id that allows us to track movement over space and time. In the event a

time field is not available, some other field that is derived from a sequential ordering may be used to a lesser effect. For example, Panoramio data always appears to have an increasing picture ID, which can be used to sort the data. However, without a precise time, some of the important filter criteria cannot be computed (such as velocity). An example set of data is shown in Listing 1 for UserIDs in the vicinity of Paris is shown below:

Listing 1: Sample Data

```
UserID,Latitude,Longitude,DateTime
jlarson,48.85522,2.352855,2012-09-08 08:20:30
jlarson,48.85667,2.348944,2012-09-08 08:20:55
jlarson,48.85546,2.348235,2012-09-08 08:21:30
ekimbrel,48.85533,2.352523,2012-09-08 08:20:30
ekimbrel,48.90883,2.401495,2012-09-08 08:20:40
ekimbrel,48.85533,2.352523,2012-09-08 12:52:42
```

2.1 Path Extraction

The first step is to partition all of the data belonging to an ID together and sort each partition by the DateTime so that we have grouped all records pertaining to a particular ID. This process was done via the "distribute by" and "sort by" HiveQL statement shown in Listing 2, which allows for distributed execution of this aggregation and sort operation. This data forms a sort of travel record and shows the sequential locations where the device or person was traveling. However, the frequency of sampling can still pose a major problem if the sampling of data is too infrequent. As such, our next step is to isolate events that are close enough together in time and space that we can use a linear interpolation between those two points as an approximation of the travel vector.

Listing 2: Line Segmentation and Filtering

```
FROM(
  SELECT userid, datetime, latitude, longitude
  FROM micro_path_initial_data
  DISTRIBUTE BY userid
  SORT BY userid, datetime asc
) mapper_out

INSERT OVERWRITE TABLE micro_path_track_extract
SELECT TRANSFORM(mapper_out.userid, mapper_out.
    datetime, mapperout.latitude, mapper_out.
    longitude)
USING python extract_path_segments.py config.
    ini
AS latitude1,latitude2,longitude1,longitude2,
    datetime1,datetime2,time,distance,velocity
```

2.2 Micropath Segmentation

The next stage of processing takes pairs of sequential rows with the same ID and creates line segments annotated with filter criteria. Filters are typically built on total distance, total time, and a composite of time and distance via a velocity calculation assuming a linear interpolation. As an approximation of the distance travelled, we use the Haversine formula to get a rough approximation of relative distance between two consecutive points. This allows for culling of noisy data as we can filter out unreasonable line segments based on physical properties. e.g. if a person traveled 100 km in two seconds, it is likely bad data and will thus be ignored.

Figure 3: 84K Flickr Derived Movement Tracks in Paris

Using the same photo metadata from Flickr for Paris as shown in Figures 1 and 2, we see what these unfiltered segments look like in Figure 3. As can be seen, this visualization is not very useful due to the lack of aggregation for a given point in space.

Time is the most powerful filter as segments with very little time delta may often produce the line segments with the highest quality. Distance filters are also important as the longer distances may imply less likelihood that a straight line was taken to travel between the two points. Finally, a third filter can be created by taking a composite value of both the time and distance variables together in the form of velocity, assuming linear interpolation. Additionally, because we have velocity, we can also start to categorize modes of transportation based on relative speed. Thus, layers of airplane, vehicle, and pedestrian travel patterns can each be analyzed. For the purpose of this explanation, we will restrict our analysis to ground movement and can ignore people taking photos from planes by capping the velocity to 40 miles per hour. A much lower speed may even be more appropriate to filter out the photos taken from a moving vehicle.

Smaller movements provide higher confidence when upsampling the data through linear interpolation. The longer the time or distance between readings, the less confidence we have in our ability to trust the linear interpolation. Thus, we want to use line segments that provide the smallest time delta between both readings. Further enhancement can also be applied by using confidence weights instead of filtering segments outright. In an urban city like Paris, we had successful results even

with a strict time filter set to as low as 15 seconds between readings. In the case of the photo metadata, this would imply that a person took a photo, walked a few feet and then took another photo within the 15 second time window, thus forming a micropath. These cases pose the greatest value to the algorithm due to the high sampling rate. Viewing these micropaths in aggregate then allows for a much broader-scope analysis.

Choosing an appropriate filter setting depends heavily on the overall coverage in the region. Certain regions with lower coverage will require less stringent filtering mechanisms, while dense urban areas can tolerate very strict criteria.

Figure 4 shows an example of this filtering process given a set of six points at specified distances in both time and space. In this example, the segments that are longer that 120 seconds and segments that exceed a velocity of 40MPH are removed from the analysis. Line segments meeting this criteria are kept and passed along to the next stage of processing. Line segments are also assumed to be of linear interpolation due to the short time span required between sampling.

2.3 Tripline Blankets

The next stage of processing is to first lay down a set of grid lines over the area of interest. Each vertical or horizontal line is then compared to every incoming line segment that was not filtered out. This process means that the grid lines act as a sort of "tripline" that records every location where the data's line segment intersects the grid, much the same way a car counter records vehicle activity in an area. By placing them at even intervals on a fine grained grid, we create a tripline blanket that captures and upsamples movement data across a wide area.

When a tripline is crossed, we not only record the precise location of crossing, but also the direction of that crossing. This allows us to then begin to analyze velocity and movement patterns to see directionality effects along with the time of day.

The strategy for choosing which gridlines to use can vary depending on the resolution of the output you want along with your filtering strategy. First, the spherical shape of the earth makes it less trivial to lay down a grid of squares rather than rectangles. While you can choose to put a tripline evenly spaced every .0001 width of latitude, it is more difficult to figure out the exact longitude lines necessary to create a square box with equal height and width. This is an area for future work. For the tests shown in this paper, we simply oversampled the data to compensate for non-symmetric grid cells. Oversampling the data works as long as you sample your grid to be finer than you actually need it, because if is possible to downsample the data at a later stage. As a practical example of this, in an urban environment

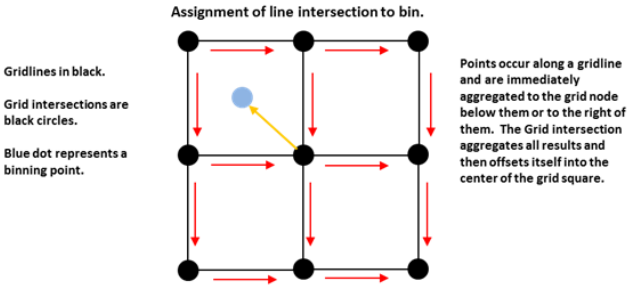


Figure 5: Grid Centering

like Paris that is significantly north of the equator, we found that running a gridline every .00001 latitude and running a gridline every .0001 longitude produced reasonable results for street level movement patterns. To further optimize the actual tripline computations, we also limit which triplines are checked for crossings with a level filter that checks the bounds of the line segment. This allows us to only check for line intersections where a tripline could actually cross.

The final results are stored with the format in the example shown in Listing 3 where the points on the initial line segment are stored, both the source and end date, the name of the grid that was applied, and finally the precise intersect location and intersect direction. This is again computed using Hive and Hadoop Streaming in concert with a Python script that performs the line intersection calculations.

Listing 3: Example Tripline Output Format

```
latitude1, longitude1, latitude2, longitude2,
    date1, date2, blanketname,
    intersectLatitude, intersectLongitude,
    intersectDirection
```

These results are then stored into a table before going through one last aggregation step, similar to the binning strategy shown at the beginning of this section.

2.4 Binning and Aggregation

The final stage of processing is simply mapping every line intersection point to the nearest grid latitude / longitude intersection as seen in Figure 5. This effectively aggregates all of the data in a given grid cell into a single point that can be displayed.

3 Discovery, Display, and Variations

After the final Grid Centering step, the results can be displayed using a logarithmic color gradient for rasterization as seen in Figure 6. Green represents the low movement flow while red represents high volume flow.

In this image, you can now start to recognize movement patterns and behaviors. In Paris, you can see several approach pathways into the area around Notre Dame. You can see how people typically walk toward

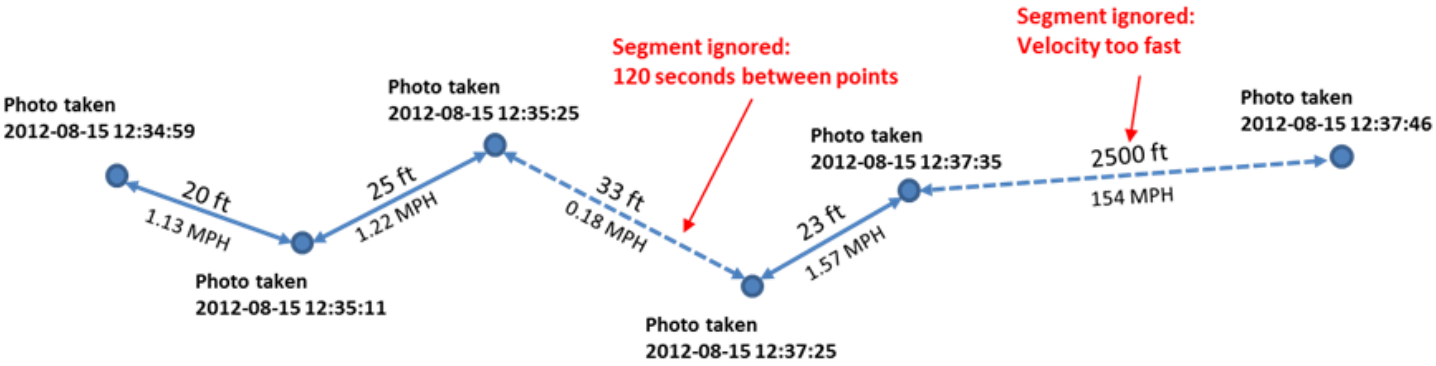


Figure 4: Line Segment Filtering

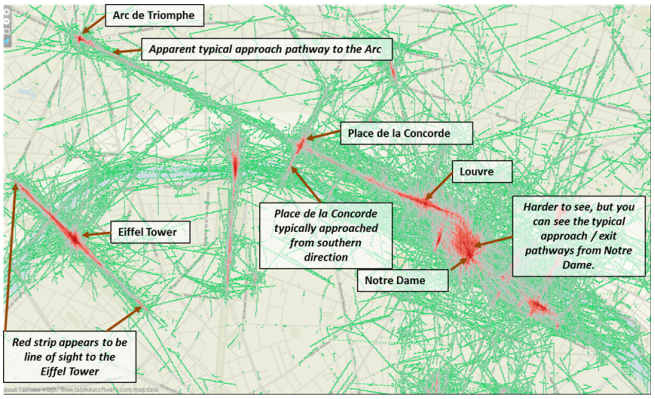


Figure 6: Final Micropaths Extracted from Paris Flickr Data

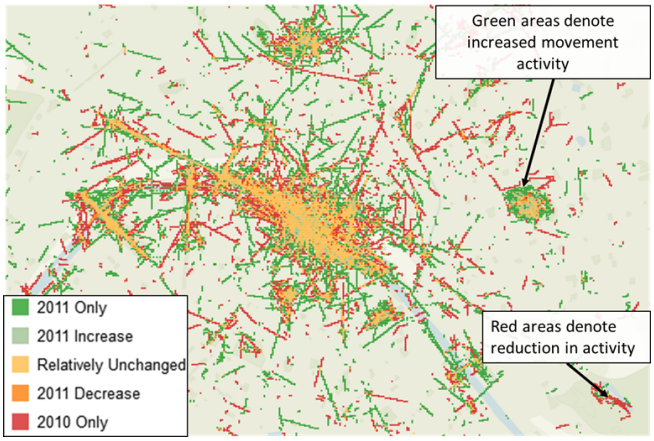


Figure 7: Flickr Aggregate Micropathing difference plot between 2010 and 2011

and away from the Eiffel tower. You can also see the north side of the Eiffel tower provides a much longer typical approach pathway on which people move - as compared to Figure 2, which shows higher raw volume (but not movement) on a long strip south of the Eiffel tower.

Other display variants include plotting the average velocity at each binning location, which gives an overall sense of the speed of normal activity within an area. This can be combined with time-of-day analysis to detect artifacts such as rush hour or other cyclical patterns that would effect velocity.

There are also several other post-processing techniques that can be applied after the tripline blankets are actually built or with slight modifications. These post-processing steps present the data over time, with direction components, at a worldwide scale, and also attempt to account for localized data collection issues.

3.1 Temporal Analysis

With the movement patterns being generated, a common next step is to analyze how those patterns change over time. Figure 7 shows an example of this on the Flickr data by comparing the change in photo activity over Paris between 2010 and 2011. In this photo, the green areas represent regions with increased movement activity, while the red regions show areas with decreased activity. To perform this analysis, first Aggregate Micropaths are generated on discrete time intervals such as year, year-month, or year-quarter. Then, each of these is compared to one another using the following pseudocode to determine significant change within an area:

```
foreach(cell in grid):
  if cell not in previous and cell not in current:
    //No data for this grid_cell
  else if cell not in previous:
    //Data ONLY exists in new readings
  else if gcell not in current:
    //Data ONLY exists in old readings
  else if log2(previous[cell]) > log2(current[cell]) + 2:
    //Substantial decrease in activity for the new reading
  else if log2(current[cell]) > log2(previous[cell]) + 2:
    //Substantial increase in activity for the new reading
```

In addition, areas where we have a reading in one dataset, but not the other allow us to plot new activity or plot cessation of activity directly. Using a scale mechanism similar to the one shown forms a baseline to ignore all the areas where activity remained relatively unchanged. However, for those regions where activity has substantially changed, this will also highlight all areas with major increases or decreases in movement activity. Binning the change levels into discrete bins is useful as a continuous gradient can be difficult to scale without interactive user feedback.

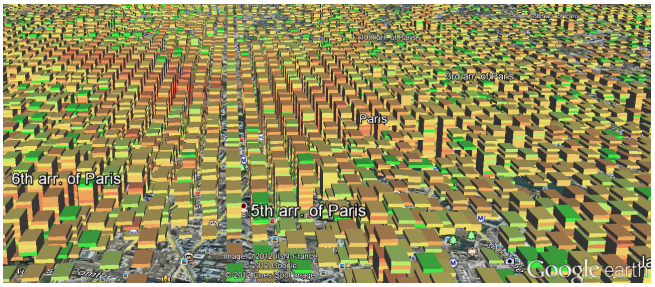


Figure 8: Paris with 24 blankets displayed on the Z-Axis. Higher Z coordinates correspond with time of day.

Figure 9: Pont de'Léna bridge with directional movement information overlaid.

Another viewpoint that is often worth capturing is the hourly movement behavior. Figure 8 shows an example of a applying Aggregate Micropathing on an hourly basis to the same Paris Flickr dataset. The approach is similar to the time extrusion mechanisms seen in [14] as the Z-axis represents the hours of the day (1-24) with the highest altitude box corresponding with hour 24 (11pm). As an example, we can see areas where people are most likely to take photos late at night (represented by the higher relief of boxes). This technique can also be used to detect major areas of rush hour by filtering out the low volume movement areas. Both of these temporal techniques can be very useful to establish base patterns of activity for the region.

3.2 Display and Direction

Generated blankets store directional information in the final output. Leveraging this directional information, we can view overall movement and flow of data. As an example, consider the visualization shown in 9, which captures movement patterns over the bridge over the Pont de'Léna near the Eiffel Tower. It is obvious to see the normal traffic patterns over the bridge, including the fact that drivers drive on the right hand side of the road.

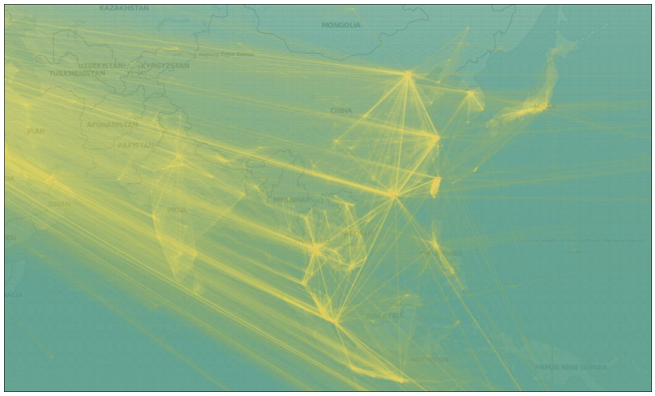


Figure 10: World View of Flickr and Panoramio data without segment filtering

3.3 World Views

While viewing small local regions can provide interesting patterns at the street level, there is also quite a bit of information to gather to understand at the world view. Using the same methodology as described in the 2 Section, we can also choose to use the tripline blanket engine to charaterize broad-scale movements as well. We do this by eliminating all filtering criteria in the Micropath Segmentation step and then using a very coarse-grained tripline blanket. For example, Figure 10 shows a worldwide view on the Flickr and Panoramio data using a coarse grid with lines every .1 latitude and .1 longitude. Immediately, patterns of travel between world cities emerge as people are using their cameras to record inter-country movement at a global scale. In particular, you can see many vivid travel patterns across Asia - based on people taking a photo in one city, flying to another city and then taking another photo upon arrival into the new city. This technique is not meant to be nearly as precise as the small area techniques described previously, but it does provide a useful overview of worldwide travel patterns.

4 Performance Measurement

Running these Aggregate Micropaths is extremely CPU intensive due to the use of the line crossing algorithm. However, the track based nature of this data make it trivial to parallelize and scale on a cluster of machines. To measure the performance and scaling ability of this technique, we ran a series of tests on a cluster of 4 machines. Each machine had 12 hyperthreaded cores, 192GB of RAM, and 24 7200 RPM HDD. Using this hardware configuration, we used different data input sizes and benchmarked the performance of the world view (no segment filtering) tripline blanket calculation at .1 x .1 latitude / longitude resolution. The input data was a 153 million row dataset that was a combination of Flickr and Panoramio photo metadata. This dataset was sampled for different data sizes and then processed

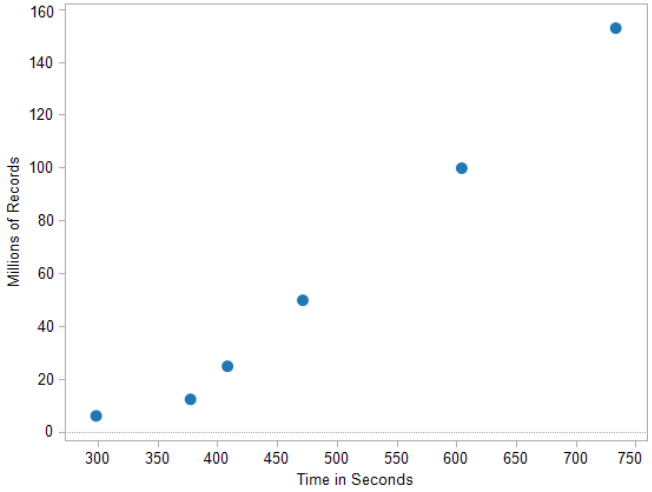


Figure 11: Benchmarks running different sized tripline blankets

through the cloud. Figure 11 shows the performance of the cloud on different sized subsets sampled from this joint dataset.

5 Future Work

The work to date has focused primarily on scalability of movement characterization using sources that contain geospatial, temporal, and ID based data. We hope to explore the automated extraction of movement types based on calculated velocities (e.g., being able to classify photos as being taken from a plane, vehicle). This may prove useful in further refining the characterization of a city's movement patterns.

Leveraging techniques that create tripline blankets wherein each grid cell represents the exact same physical area will also improve the quality of results. This will involve writing a grid generator that dynamically chooses triplines that are adjusted according to the curvature of the earth. Current mechanisms for this involve manual selection and are very inefficient. Another option would be to leverage a Geodetic System like ENU (East-North-Up) that is centered on a locla area of interest *REFERENCE HERE*. Along this same research line, investigating bin sizes and their impacts at different level of abstraction is another area for further research.

Finally, we see a need to further investigate analytics that focus on the direction of movement in aggregate.

6 Acknowledgements

Many people have been involved in this research and contributed to this paper. Special thanks to Dr. Chris White who sponsored this work and provided technical guidance that greatly improved the quality of results. Galen Pickard helped define the original individual tripline technique with directionality. Thanks also to all the reviewers, Tom Bougan, Al Reich, Eric Dull, and others who helped organize and refine the paper's message.

References

[1] AS Pentland, *"Honest Signals"*. The MIT Press, 2008.

[2] Yahoo, *"Flickr"*. www.flickr.com, 2012.

[3] Google, *"Panoramio"*. www.panoramio.com, 2012.

[4] Twitter, *"Twitter"*. www.twitter.com, 2012.

[5] OpenStreetMaps, *"OpenStreetMaps"*. http://www.openstreetmap.org, 2012.

[6] FourSquare, *"FourSquare"*. http://www.foursquare.com, 2012.

[7] Wikipedia, *"Automatic Identification System"*. `http://en.wikipedia.org/wiki/Automatic_Identific` 2012.

[8] Santi Phithakkitnukoon et. al., *"Taxi-Aware Map: Identifying and Predicting Vacant Taxis in the City"*. SpringerLink, 2010.

[9] Justin Cranshaw et. al., *"The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City"*. AAAI, 2012.

[10] Matei Zaharia et. al., *"Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing"*. http://www.spark-project.org/, 2012.

[11] Tominski *"Tominski paper here"*. Infovis, 2012.

[12] Eric Fisher http://www.flickr.com/photos/walkingsf/sets 2012.

[13] Slava Kisilevich, Milos Krstajic, Daniel Keim, Natalia Andrienko, Gennady Andrienko. *"Event-based analysis of people's activities and behavior using Flickr and Panoramio geotagged photo collections"*. 2010

[14] M.P. Kwan and J. Lee. *"Geovisualization of human activity patterns using 3D GIS: a time-geographic approach"*. Spatially integrated social science, 27, 2004

[15] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. *"Trajectory pattern mining"*. In 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 339, 2007

[16] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, Gennady Andrienko. *"Visually-driven analysis of movement data by progressive clustering"*. Information Visualization, 7, 3(4):225-239, 2008

[17] Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma. *"Mining interesting locations and travel sequences from GPS trajectories"*. In Proceedings of the 18th international conference on world wide web, pages 791-800. ACM New York, NY, USA 2009