Visualizing Geotagged Twitter Images during Disaster Events

Jennings Anderson Melissa Bica

Department of Computer Science
University of Colorado Boulder

Abstract

Visualization is a useful way for analysts to gain insight into their data. Unfortunately, visual data is often excluded from visualization due to the challenges faced in dealing with images, such as storage and processing. We have designed and built an interactive viewing tool that centers around visual data. Our tool, TweetsOnAMap.com™, visualizes geotagged and geolocated image tweets and allows for exploration of the data from multiple perspectives. This visualization tool is intended for researchers to analyze visual social media data, and as such we conducted preliminary evaluation with researchers. Users found our tool intuitive and engaging and also provided valuable feedback for future design improvements.

1 Introduction

There are plenty of systems to visualize textual information from online social media, but visualizing images is more challenging. Many social media posts contain images that are critical to incorporate in social media analysis, but because of the difficulty in visualizing such large amounts of visual data, they often get excluded from both the visualization and the resultant analysis.

In our crisis informatics research, we often analyze Twitter data collected during disaster events which include both textual and visual information, as well as temporal and geographic attributes. One common way of analyzing such data is to upload it to a database and filter for specific attributes relevant to the research questions. This lends itself well to text processing, such as searching for tweets containing “Frankenstorm or “Hurricane Matthew, as well as filtering based on geographic and temporal boundaries. Many tweets also contain multimedia information such as images, videos, and gifs, which have been shown to have higher user engagement than tweets containing only text [5]. However, this content is not readily viewable nor able to be processed and filtered in a database the way that textual information is. If only using a textual database for data analysis, much information is thus excluded.

A first step to addressing this problem is to visualize these data. Histograms and other basic charts are often useful to view trends in a dataset over time or in aggregate, while maps are useful for viewing geospatial patterns. Images are obviously best interpreted visually; unfortunately, however, they are difficult to sort through. Combining the images with the geographic and temporal metadata, we created a visulization tool that allows for spatiotemporal filtering of these large image datasets.

2 Related Work

Image visualization tools and approaches such as JustClick [2] and Fan et al.’s supervised classification method [3] allow filtering and clustering based on image characteristics as determined by automated and semi-automated means. These methods allow organization of the images by similar attributes. Our tool instead organizes the images spatio-temporally, in terms of from where and when they were posted. This is a similar approach to the interface available on Flickr.com to view geotagged images.

Santos et al. [6] also presented a method for visualizing image data and retrieving images based on content, or visual features. While we do not implement image feature extraction, we share a similar goal in terms of interaction: allowing the user to take a more “active role” in exploring the visualized data by querying and filtering based on their own research needs as well as their prior knowledge of the data or its context. In the future we intend to extract image features as described in this work, and then allow users to filter and query the data based on these features as well.

While our tool does not currently enable querying and filtering based on learned classification, our choice to use mapbox-gl as the technical back-end allows us to selectively render images based on any textual property of the geojson feature representing each image. Future iterations of our tool will be sure to incorporate such capabilities.

In terms of how this visualization tool could be used by researchers, we see its main purpose as a tool to help create insight [4]. It is not meant to simply put data onto a map, but to allow researchers to explore the data from many perspectives and find new meaning that they would not have been able to without such a visualization. We attempted to informally measure how well our visualization tool achieves this purpose by conducting a qualitative user evaluation as suggested by North, discussed in Section 4.

3 Implementation

3.1 Overview of the Tool

TweetsOnAMap.com consists of a primary map visualization and a side pane for displaying images. The map has two filtering options, to *Show **geotagged** tweets* (tweets that are tagged at a specific point (an exact latitude and longitude) and to *Show **geolocated** tweets* (tweets that claim to be posted from a broader location, such as a city or country). Geotagged tweets are represented as orange points, and geolocated tweets are represented as blue clusters around the approximate center of the specified location. For instance, a large circle over Cuba on the map represents all the tweets geolocated in “Cuba.” There is an additional display option to *Show geotagged images on map*, which shows a small preview of a tweet’s image at the specific geographic point at which it was tagged, remaining at that point as a user zooms or pans around the map. This feature is not appropriate for geolocated tweets, as their exact location is unknown and is therefore not available.

The map also contains a histogram of tweets per day which serves two purposes: 1) to show the distribution of all tweets across the dates of collection for the specific event and 2) to filter the map based on a date range. To filter, a user can click and drag a box around the days of interest, and the map and images update to reflect only tweets posted within that duration.

Finally, the side image pane displays the images included in all the tweets in the current map view. When a specific image is selected, a larger version of the image appears above the panel along with additional tweet metadata, including the text, date it was tweeted, and Twitter handle. The image itself links to the original tweet on Twitter to see its original context. The clickable Twitter handle serves as a way to filter the visualizations to only display tweets by

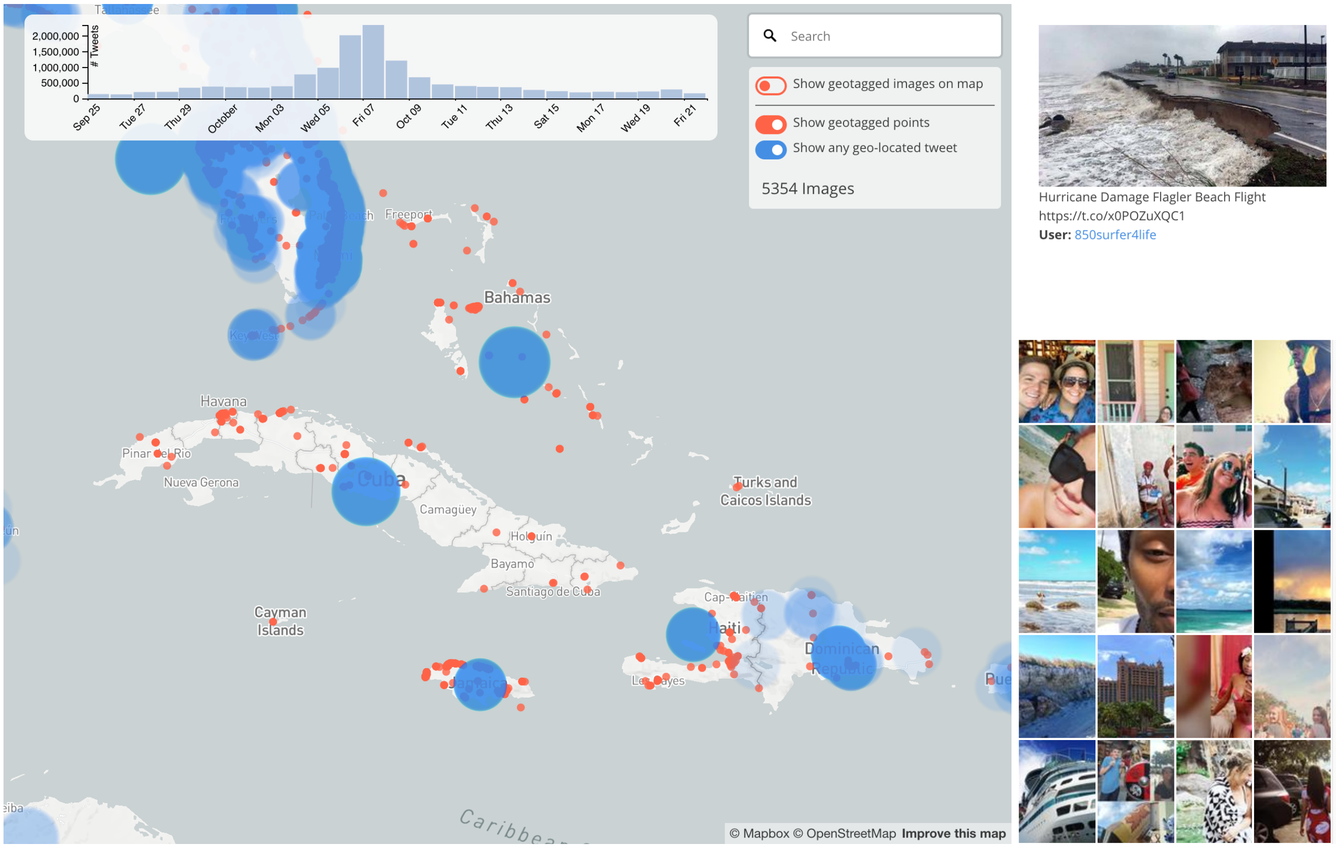

Figure 1: Tweetsonamap.com interface.

the selected twitterer. Finally, hovering over any image within the image pane will highlight the geotag point on the map at which that image was tweeted to provide geospatial context.

3.2 Preprocessing

An important requirement for our visualization tool is the ability to adapt to new events as they occur. Using Jupyter with both Python and NodeJS notebooks, we designed a pipeline to 1) query Twitter (GNIP) for tweets with both images and geographic metadata (geolocated or a geo-tag), 2) download the resulting tweets, 3) download and scale the attached images, and 4) generate geojson feature collections from the tweets. The resulting feature collection maintains only the coordinates, id, text, date, and username of each tweet object, creating the smallest possible object while preserving some of the most essential aspects of the tweets.

3.3 The Map

The visualization is built atop Mapbox-GL, an open source JavaScript web mapping library designed to efficiently visualize vector geospatial datasets (built on WebGL). The API easily ingests geojson, turning it into vector-tiles for efficient processing and display. The map then becomes the main control object for the page. Each time the map is moved (and fires a `moveend` event), the map queries each layer of tweets (geolocated and geotagged) for currently rendered features: those visible in the current viewport.

Other filters, such as date range or user are also handled by the map. This optimizes rendering and map performance. Filters are applied to each layer, preventing the objects from being rendered and thereby excluding them from the list of visible features.

3.4 Image Pane

Once a list of visible objects is created, it is passed into an image rendering function that first clears all present items and then adds list items to the image pane with references to the tweet thumbnail image, as determined by the tweet id. At first, only enough images are rendered to fill the page with the rest of the tweet ids preserved in memory. If the user scrolls inside the image pane, the remaining tweets are rendered from the list in memory.

3.5 Extensibility

A major design goal was the ability to easily import new datasets for quick visualization. For this, we implement visualization controllers for each piece of the page: the timeline, the image pane, and one for each type of object on the map. On page load, the data sources for each of these objects is read from a separate configuration file or URL variable, enabling a user to load any (pre-processed) dataset. Our pre-processing pipeline is able to process about two tweets per second: the bulk of this time consumed downloading and rescaling individual images.

4 Design Choices

We employed several design choices based on information visualization techniques. First, we represented the uncertainty of geolocated tweets by blurring the edges of the blue circles representing clusters of tweets at a location. This blurring helps to indicate that the geospatial boundaries of the cluster and the circular shape are not precise, but approximate [1].

Second, we implemented several different filtering techniques to follow Shneiderman's "Visual Information Seeking Mantra": "Overview first, zoom and filter, then details-on-demand" [7]. Our map visualization provides the main overview by showing the geographic distribution of tweets (as well as the images attached to those tweets if the *Show geotagged images on map* is selected). The user can pan and rotate the map to explore this overview of the data. Next, the user can zoom in to a particular geographic region to see more precise locations of tweets, as well as filter the tweets visible on the map. Filters include temporal and level of geospatial detail (i.e.ġeotagged or geolocated tweets). Finally, details-on-demand functionality is implemented in the orange geotagged tweet points on the map. Clicking on these brings up the image in the side image pane as well as other details of the tweet. Additionally, clicking on any image within the side image pane itself also brings up these details-on-demand. We opted against having a pop-up when a user clicks on a tweet on the map visualization as to avoid too much information in one place. The side image pane provides a separate area dedicated to viewing details-on-demand.

The timeline visualization uses a brushing visualization technique for a user to select a range of days and display only tweets and images posted within that range. The brushing effect allows the user to easily select a continuous range and have that range clearly marked. The brush area also snaps to the boundaries of each bar on the graph so that only full days can be selected.

Finally, we considered the use of color in our design. We chose distinct colors for geotagged versus geolocated tweets, and colored the toggle switches to match these so that the actions of these switches is very intuitive. We colored the switch for *Show geotagged images on map* the same as for *Show geotagged points* as these both act upon geotagged, not geolocated, tweets. We separated these two orange switches to distinguish that one is a display option while the other is a filtering option.

5 Findings & Discussion

To evaluate our system, we performed a user-evaluation and cognitive walk-through with a prominent crisis informatics researcher. Using the map as the main control seemed non-intuitive at first, but the researcher quickly caught on and once they were comfortable with the common map interactions such as zooming and panning, they understood how the images present in the image pane could be controlled by the location of the map. Filtering the visible features with the toggle switches and the timeline were straightforward tasks.

The researcher was confused that though clicking on the geotagged tweets performed an action (loading the image in the image pane), clicking on the geolocated tweets did not. Initially, this was an intentional design decision because the blue circles representing geolocated tweets do not represent a single tweet but rather an area of uncertainty where multiple tweets may be located. After a short discussion with the researcher, we opted to enable a `flyTo` action when clicking on a cluster of geolocated tweets. This action zooms and pans the map to the area clicked, invoking the map's native filtering action of loading the visible tweets in the image pane. In this manner we are not prioritizing one image over another, but still enabling an intuitive map action: clicking on an area of the map renders the images posted in that area in the image pane.

Finally, in conducting this user evaluation, we were not only evaluating the usefulness and intuitiveness of the visualization tool, but learning about the context in which this tool would be used in a real-life research setting [8]. This helped us to contextualize our design decisions around our tool.

References

[1] K. Brodlie, R. A. Osorio, and A. Lopes. A review of uncertainty in data visualization. In *Expanding the frontiers of visual analytics and visualization*, pp. 81–109. Springer, 2012.

[2] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li. JustClick: Personalized image recommendation via exploratory search from large-scale Flickr images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):273–288, 2009. doi: 10.1109/TCSVT.2008.2009258

[3] J. Fan, W. Ribarsky, and R. Jain. Social computing for collaborative image understanding. *Invited Paper, IEEE Multimedia*, pp. 1–15, 2011.

[4] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006. doi: 10.1109/MCG.2006.70

[5] S. Rogers. What fuels a Tweet's engagement?, 2014.

[6] L. F. Santos, R. L. Dias, M. X. Ribeiro, A. J. Traina, and C. Traina. Combining diversity queries and visual mining to improve content-based image retrieval systems: the divi method. In *Multimedia (ISM), 2015 IEEE International Symposium on*, pp. 357–362. IEEE, 2015.

[7] B. Shneiderman. The eyes have it: a task by data type taxonomy for informatio nvisualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996. doi: 10.1109/VL.1996.545307

[8] M. Tory. User Studies in Visualization: A Reflection on Methods. In *Handbook of Human Centric Visualization*, pp. 551–580. 2014. doi: 10.1007/978-1-4614-7485-2