

# CityBeat: Real-time Social Media Visualization of Hyper-local City Data

Chaolun Xia  
Rutgers University  
New Brunswick, NJ, USA  
xia.chaolun@gmail.com

Raz Schwartz  
Cornell Tech  
New York, NY, USA  
raz.schwartz@cornell.edu

Ke Xie  
Cornell Tech  
New York, NY, USA  
kx29@cornell.edu

Adam Krebs  
New York University  
New York, NY, USA  
krebsadam@gmail.com

Andrew Langdon  
Columbia University  
New York, NY, USA  
andrewlngdn@gmail.com

Jeremy Ting  
Rutgers University  
New Brunswick, NJ, USA  
jting0193@gmail.com

Mor Naaman  
Cornell Tech  
New York, NY, USA  
mor.naaman@cornell.edu

## ABSTRACT

With the increasing volume of location-annotated content from various social media platforms like Twitter, Instagram and Foursquare, we now have real-time access to people's daily documentation of local activities, interests and attention. In this demo paper, we present *CityBeat*<sup>1</sup>, a real-time visualization of hyper-local social media content for cities. The main objective of *CityBeat* is to provide users – with a specific focus on journalists – with information about the city's ongoing, and alert them to unusual activities. The system collects a stream of geo-tagged photos as input, uses time series analysis and classification techniques to detect hyper-local events, and compute trends and statistics. The demo includes a visualization of this information that is designed to be installed on a large-screen in a newsroom, as an ambient display.

## Categories and Subject Descriptors

Information systems [Database management]: Database applications—*Data mining*

## Keywords

Event detection, Data mining, Social media, Visualization

<sup>1</sup>The live system as well as a video demonstration are available at <http://bit.ly/citybeatwww14>

## 1. INTRODUCTION

Users of social media platforms such as Instagram and Twitter produce large volumes of geo-tagged content and publicly share in real time. These photos, videos, check-ins and tweets are used to document people's interests, opinions and in some cases, what is happening around them.

The goal of *CityBeat* is to provide users – in particular, metro-area journalists – with an ambient real-time visualization of social media activities in the city. This visualization is based on multiple streams of data from social media platforms, like Twitter, Foursquare and Instagram. One key feature of *CityBeat* is hyper-local event detection. While previous work mostly focused on large-scale, e.g. global or national level, event detection [2, 4, 6], *CityBeat* focuses on *hyper-local* events, that are occurring in a small region, e.g. a street corner or a certain venue rather than city- or country-level area. These events can range from music concerts and exhibitions to emergencies like fires and car accidents. Performing robust hyper-local event detection is challenging, given the noise and scale of social media data, where the sparse signal for any location can be easily overwhelmed by irrelevant content. Furthermore, as finding out and following breaking events in real-time is highly important for our users (journalists and city officials), our system has to focus not only on robust and precise detection but also on efficient detection in real time [7].

To address these challenges, our demo system models the time series of location-based streams of geo-tagged photos, and detects abnormal signals that suggest candidate events in a hyper-local regions. Each candidate event (i.e. abnormal time series signal) is then represented in the feature space of **S**patial, **M**eta, **T**extual and **H**istorical (SMTH) features, and classified as a true event or noise (a detailed investigation of the initial implementation of the algorithm is available in [10]). Moreover, we incorporate crowdsourcing input to refine our classification results. Finally, the particular social media posts that are part of the detected event are ranked according to their relevance and then visualized in an ambient display.

In addition to detecting events, our *CityBeat* visualization includes a number of other elements reflecting the city’s social media activity, including activity volumes and their divergence from normal (predicted volume) and other information components, as detailed below.

The rest of this paper is structured as follows. In Section 2, we briefly review previous research on event detection in social media. In Section 3, we introduce our *CityBeat* visualization. In Section 4, we review our event detection system. In Section 5, we describe the proposed demonstration plan.

## 2. RELATED WORK

A number of recent papers have examined hyper-local event detection. Closest to our work is the Whoo.ly system [3], which discovers and summarizes hyper-local information about events, places, topics and people from Twitter content to serve hyper-local communities. Our system has some overlaps with (and borrows some ideas from) Whoo.ly, but our data (Instagram-focused), methods, and, more than anything, visualization goals and techniques are quite different.

Focusing on textual content, prior work [9] represents and detects events by grouping a few *significant terms* whose frequency of use in geo-tagged Twitter content changes significantly during a short period. In the framework proposed by [8], tweets are incrementally grouped into clusters based on their locations, and then filtered by candidate classifier.

Our event detection component of *CityBeat*, initially presented in [10], has several commonalities with these projects. However, a number of key differences exist. First, we focus on Instagram photos which contain both text and image content for analysis and visualization. Second, our we present a unique event detection approach, combining time series and classification methods based on SMTH features. Furthermore, we incorporate crowdsourcing to improve classification and overall system precision. Finally, we focus on incorporating event detection output into a real-life, working visualization that is used by journalists and city officials.

## 3. USER INTERFACE

*CityBeat* is a web-based ambient visualization meant to be shown on large displays in a newsroom or similar setting (Fig 1). In this visualization, we show our detected events as they are extracted from Instagram as well as real-time statistics extracted from Instagram, Twitter and Foursquare. We have 4 main components on the user interface: the Detected Events List, Event Window, Statistics Sidebar and Trending Venues Ticker.

As shown in Fig 1, the *Detected Events* List is an auto-scrolling list of all the events discovered in the past 24 hours. Each detected event on the list is noted on the background map by a pulsing circle placed in a specific geographical location. As the map moves from one location to another at the center of the screen, an *Event Window* presents a summarization of the currently in-focus event including keywords, time of detection and relevant photos, all automatically computed by our system. Clicking on an Event Window directs the viewer to an event page (shown in Fig 2) in which all of the event information is presented.

The *Statistics Sidebar* and *Trending Venues Bar* provide live data such as the number of tweets and photos per hour,

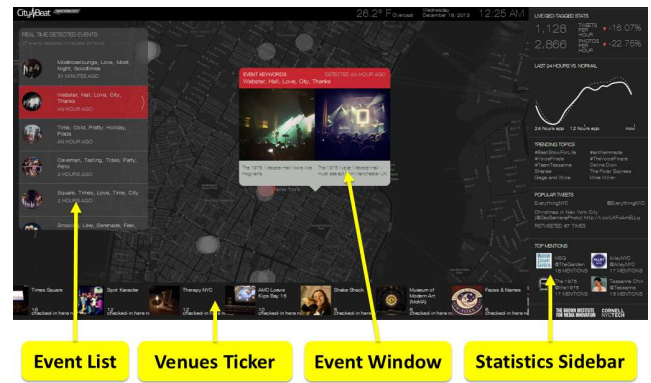


Figure 1: Screen shot of *CityBeat* ambient display interface with 4 components: Detected Events List, Event Window, Statistics Sidebar and Trending Venues Ticker.

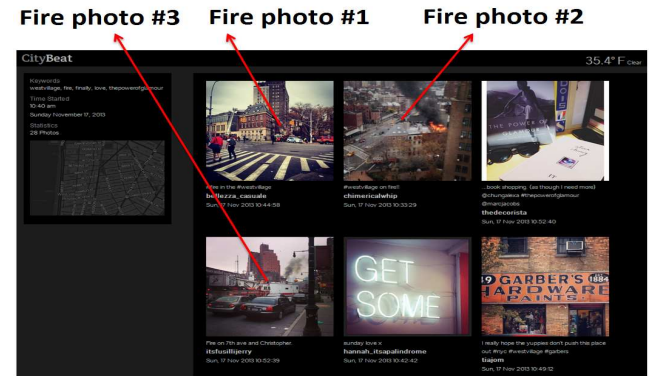


Figure 2: As an example, this event page shows the photos of the fire detected on Nov 17, 2013 at West Village.

top-mentioned twitter accounts, top-retweeted tweets, Twitter’s trending topics for the city, and Foursquare’s trending venues. In the sidebar, we also plot the time series of photos and tweets volume for the city during the past 24 hours. The actual data from the past 24 hours are visualized by a solid colored curve, while the dashed curve represents the predicted time series (expected, “normal” levels). These two curves can provide a quick visual indication when some unusual city-wide activity is occurring. Below the volume graph, we show the city’s top 10 trending topics extracted from Twitter APIs, the five most retweeted city tweets, and the top four handles who got most mentions during the past hour by users in the city. Again, these elements intend to give the users a quick idea of what is the “social media state” of the city, even if these are not necessarily related to breaking events. At the bottom of the user interface, the Trending Venues Bar presents the top checked-in Foursquare venues within the city. We are extracting the name, image and check-in count of a venue directly via the Foursquare APIs.

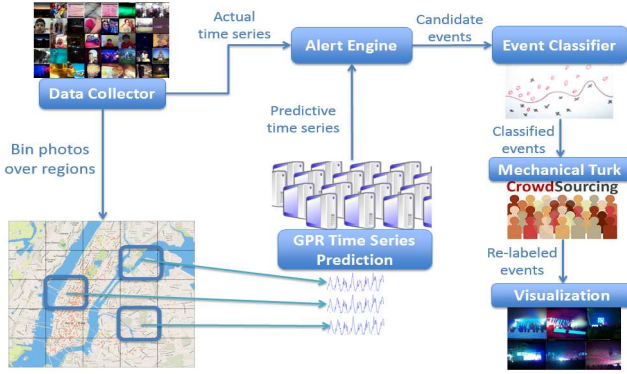


Figure 3: Architecture of CityBeat.

## 4. EVENT DETECTION SYSTEM

Our event detection system consists of six components: photo stream crawling, time series prediction, online alarm engine, event classification, relevant photo ranking, and mechanical turk re-labeling. The work flow between these components is shown in Fig 3.

Our crawler constantly collects geo-tagged photos through Instagram’s search APIs, and stores the content in a database. The time series prediction component takes the historical photos as input and models photo volume time series for each region. As output, it generates the predictive time series for the next day in that region. The online alert engine compares the prediction with the real-time data from the crawler. If it discovers an abnormal signal, i.e. a significant deviation, a *candidate event* is created, containing all the photos that contribute to the abnormal signal. In the next step, the candidate event is transformed into a vector of SMTH features, then passed into the event classifier that determines whether the candidate event is a true event by supervised learning technique. We also incorporate crowd-sourcing to improve the classification precision by removing false alarms. Finally, all the true events are visualized on the user interface after ranking the photos by their event-relevant score.

### 4.1 Geo-tagged Photo Stream

Instagram recently announced that it has 150 million users who are posting 55 million photos daily<sup>2</sup>. In this work, we only focus on geo-tagged photos taken within New York city limits. Our crawler collect geo-tagged photos through the Media Search APIs<sup>3</sup>. During the month of January 2014, our system collected on average 94,618 photos per day. Although in this paper we focus on the area of New York City, this system can be generalized to any city or country, or to other geo-tagged social media data source. In fact, we also crawl Twitter data geo-tagged to New York to use in the various components of *CityBeat*.

### 4.2 Time Series Prediction

Before detecting an abnormal signal from time series, we create a predictive model, as described in detail in our previous work [10]. Given a region  $r$ , we use the data from the

last 14 days to create a time series of photo volume. We use Gaussian Process Regression (GPR) with periodical kernels to model the predictive time series.

We split New York into 873 non-overlap rectangular regions, with an average size of 590 meters by 675 meters. We use GPR to train and predict the time series for the next day in 188 of these regions with high enough volume of photos. For the 685 remaining regions, we use a simpler and more computationally-efficient time series modeling method which takes the average of past periodical observation as prediction. The time series computation is running on a cluster of distributed machines.

### 4.3 Alarm Engine

Based on the assumption that a sudden increase in the number of photos is a potential indication of an event, we compare our predictive time series generated by time series prediction component in 4.2 with the actual time series generated by the real-time crawler. If within the time period  $[t, t + \Delta t]$ , the number of photos in a region  $r$  is obviously larger ( $zscore > 3$ ) than its predicted value<sup>4</sup>, the alarm engine will instantly create an alarm, also called candidate event, by retrieving all the geo-tagged photos that were taken in  $r$  during  $[t, t + \Delta t]$ .

### 4.4 Event Classification

As described in 4.3, the online alarm engine outputs candidate events. However, we find that only a small portion (13.3%) of all the candidate events are true events. We built a binary SVM classifier that takes a candidate event as input and outputs a label that indicates whether the input candidate event is a true event or non-event. The classification is based on 22 SMTH event features in four categories that are Spatial, Meta, Textual and Historical features. The complete feature list and their interpretations are available at [10].

The training and evaluation data are labeled by crowd-sourcing workers and journalists from a local newsroom in NYC. The overall test results on balanced data (with the same number of positive and negative data points) show that the precision, recall and accuracy are all above 85%. The results on real unbalanced data where the percentages of positive and negative data are 13.3% and 86.7% respectively reach 50% precision, 87% recall and 88% accuracy. The drop of precision is mainly caused by the unbalanced data where the negative data dominate. To tackle with unbalanced data problem [5], we calibrated the classifier by imposing a tuned cost matrix. The detailed experimental results are available at [10].

### 4.5 Relevant Photo Ranking

For each event, only some of the photos in the event’s geo region and time span would be relevant. To highlight the relevant information in the *CityBeat* presentation, we rank the photos based on their relevance to the detected event. The normalized relevance score of each photo to the event is the linear combination of the textual similarity and the geographical closeness between the photo and the centroid [1] of the event.

<sup>2</sup><http://blog.bufferapp.com/instagram-stats-instagram-tips>

<sup>3</sup><http://instagram.com/developer/endpoints/media>

<sup>4</sup>For each prediction, GPR provides predictive standard deviation that we use to define and compute *zscore*.

## 4.6 Mechanical Turk Re-labeling

Although the classifier described in 4.4 normally achieves good accuracy, we have added a “thin” layer of manual annotation to provide more accurate results (and allow the algorithm to be tuned for higher recall with some precision cost that is fixed by the Turkers). Once the classifier detects a true event, our system sends the event to Amazon Mechanical Turk using its APIs. The Turk workers give their judgment on the classified event as well as the event-related photos, in multiple independent steps. Each classified event is re-labeled by 3 Turk workers, and we take the majority of votes as its finalized label. This re-labeling process is automated. In most cases, the Turk workers can complete re-labeling for an event within one minute of the event detection time.

## 5. DEMONSTRATION

Our *CityBeat* demonstration visualizes real-time social media data from New York City, as shown on <http://bit.ly/citybeatwww14>. *CityBeat* presents constantly updated real-time information for New York City, including detected events within the last 24 hours, as well as the other information as shown in Figure 1 such as time series of Instagram photos and tweets, and statistics (top tweets, venues, topics and mentions). The information updates every three minutes with the latest events and statistics. Focusing on a highlighted event, a user can learn about the event’s topic using the extracted keywords and its geographical location by tracing the red pulsing circle placed on the city map. To learn more about an event, users can click on the Event Window and be directed to the corresponding event page showcasing all the photos and details from that specific event.

Already running over several months, our system detected various types of events including fires, car accidents, parades, concerts, sport games etc. For example, on November 17th, 2013, a fire started in a building in the West Village neighborhood of New York City. This fire was reported by several local blogs. Our system detected the fire, as shown in Figure 2. From the event page, we can see that the top keywords for the event included *westvillage* and *fire*. The center of the event’s page map is the estimated location of the fire, correctly identified as the West Village. The time when the fire was detected is shown as 10:40am; to the best of our knowledge, the system detected and reported the fire hours before any other local news outlet reported on it. Shown in Fig 2, the top 2 photos were taken from the site of the fire.

## 6. CONCLUSION AND FUTURE WORK

In this demo paper we present *CityBeat*, a real-time social media visualization for cities, robustly detecting hyper-local events and providing live city statistics. Our system helps our target users, journalists and city officials, to follow the rhythm and activities of the city by detecting and visualizing the events around them and the social media pulse of the city.

Future work should extend the framework and integrate multiple data sources, e.g. Twitter data, to detect and categorize hyper-local events automatically. To improve the classification and ranking of content, future work could consider using computer vision techniques to analyze image and multimedia content. Finally, as we are continuously collecting re-labeling feedback from crowdsourcing, a future step

could incorporate the feedback in real time to improve the classification model and accuracy. Finally, we are currently developing the desktop and mobile experiences around the *CityBeat* data, which will allow for additional investigative tools and location-driven alerts, respectively.

## 7. ACKNOWLEDGMENTS

This work is supported by a Magic Grant from the Brown Institute for Media Innovation, and the National Science Foundation grants Numbers 1054177 and 1017845. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 8. REFERENCES

- [1] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the 13rd ACM International Conference on Web Search and Data Mining*, pages 291–300, 2010.
- [2] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 523–532, 2009.
- [3] Y. Hu, S. D. Farnham, and A. Monroy-Hernández. Whoo.ly: Facilitating information seeking for hyperlocal communities using social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3481–3490, 2013.
- [4] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the International Conference on Management of Data*, pages 1155–1158, 2010.
- [5] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
- [6] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860, 2010.
- [7] R. Schwartz, M. Naaman, and Z. Matni. Making sense of cities using social media: Requirements for hyper-local data aggregation tools. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2013.
- [8] M. Walther and M. Kaisser. Geo-spatial event detection in the twitter stream. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, pages 356–367, 2013.
- [9] A. Weiler, M. H. Scholl, F. Wanner, and C. Rohrdantz. Event identification for local areas using social media streaming data. In *Proceedings of the 3rd Workshop on Databases and Social Networks in SIGMOD*, pages 1–6, 2013.
- [10] K. Xie, C. Xia, N. Grinberg, R. Schwartz, and M. Naaman. Robust detection of hyper-local events from geotagged social media data. In *Proceedings of the 13th Workshop on Multimedia Data Mining in KDD*, 2013.