Crowd Sourced Data Collection and Analysis

Manish Raj
Department of Computer Engineering, GHRCEM
DomkhelRoad , Gate No.:1200
Wagholi, Pune – 412207, MH – India

Rohit Yadav
Department of Computer Engineering, GHRCEM
DomkhelRoad , Gate No.:1200
Wagholi, Pune – 412207, MH - India

Seema Udrikar
Department of Computer Engineering, GHRCEM
DomkhelRoad , Gate No.:1200
Wagholi, Pune – 412207, MH - India

ABSTRACT

Today's smartphones can unlock the full potential of crowd sourcing and take eParticipation to a new level. Users can transparently contribute to complex and novel problem solving solutions. Engagement of citizens is still challenging but the proliferation of smartphones with geolocation has made it easier than before. The paper introduces the environmental project and associated mobile app called CROWDPATROL. The project is primarily intended to report illegal waste dump sites in the district of Pune, Maharashtra. The idea is to use the potential awareness of the broader public about the environmental and economic drawbacks of illegal landfills. The smartphones/tablets GIS (geographic information system) reporting application CROWDPATROL has been developed.

The mobile app "CROWDPATROL" enables users to report illegal dump sites, potholes and various other civic issues. The project CROWDPATROL is intended for all the citizen who would like to be actively involved in reporting, mapping and tracking of illegal dump sites in their cities or villages originating mainly because of bad attitude and behaviour of irresponsible people and/or businessmen.

The objective of this project is to contribute to solving problem of environmental pollution by illegal dumps in Pune and contribute to "SWACH BHARAT ABHIYAN"

General Terms

GPS, GIS, Crowdsourcing, Heatmaps, Google Maps API, Geo-tagging.

Keywords

CrowdPatrol, Crowd patrol, Swatchh Pune

1. INTRODUCTION

Collection of massive amounts of data is one of the most labour intensive tasks in any industry. Various government bodies associated with civic duties such as transport department, municipal corporations are often burdened with the task for surveying before mitigating issues. The system proposed in the paper exploits crowd sourced architecture to collect data and report issues to various civic bodies in real time. Citizens are provided with a platform where they can report various issues using the app. The app runs on a GPS enabled smart phone and uses goe-location to find user's location, camera sensors to capture associated images which can then be submitted to a central cloud based infrastructure. The app is primarily designed to runs on android platform with API level 15 and above which targets more than 88% of available smart phones running Android.

2. LITERATURE SURVEY

In a paper "crowd sourced approach for mapping of illegal dump sites in Czech republic"1] by MiroslavKubasek, he demonstrated the project that maps geo-spatial data on an interactive map. He worked on solving the problem of illegal dumps in Czech Republic by utilizing the power of crowd.

Another author Mr. Christian Heipke in his paper "Crowdsourcing geospatial data"[2] exploited the concept of collection and analysis of geo-spatial data.

Also in India, the paper "A social Incentive System to Crowd source Road Traffic Information in Developing Regions"[3] by RijurekhaSen also highlighted the power of crowd sourcing and demonstrated how developing a crowd sourced application can help in designing real time applications for monitoring traffic congestion on various routes within a city.

Apart from studying various academic paper, we studied various concepts and surveyed different methodologies which can be exploited for successful design and implementation of the project.

GEOTAGGING[6]:
Geo-tagging is the process of adding geographical information to various media in the form of meta-data. The data usually consists of coordinates like latitude and longitude, but may even include bearing, altitude, distance and place names. Geo-tagging is most commonly used for photographs and can help people get a lot of specific information about where the picture was taken or the exact location of a friend who logged on to a service. Making and preserving geographical associations with pictures is an age-old process. During the ―film-camera‖ days, people would write the place where the picture was taken on the back of the print. Today, a user can map his pictures precisely using systems such as Google Picasa™, Google Earth, and Yahoo Flickr. With the massive volume of digital imagery being captured and shared on the Web, and the phenomenon of geotagging having acquired phenomenal proportions, it has become a recent research trend to explore computer vision algorithms to link user-tags, visual content of pictures, and community knowledge with the geographic locations where the pictures were captured. An important research question that motivates the current work is how this massive volume of community geotagged image data can be leveraged for assigning geographic locations to images, especially legacy pictures that were taken before cameras could interface directly with GPS receivers.

In an automatic geotagging algorithm based on simple K-nearest-neighbor visual search to infer geo-association of images was described. The basic premise explored is that the visual content of pictures and their geographic locations are correlated.

The strength of the system lay in a simple technique and the availability of a very large-scale image database (~6 million images) for search. In the recent work, we built upon and studied the question of incorporating both tag annotations in addition to K-nearest-neighbor visual search to refine the geo-inference.

To this end, we analyses geographic distributions associated with tags with standard tools to examine information content: frequency, and mutual information. Experiments have shown that user annotations alone can be used to infer the location of pictures accurately.

We show that the geographic probability distribution of a tag relates to the semantic meaning of the tag itself and we can effectively determine which tags are cities or nations by examining the tag maps themselves. Eventually, we expect better fusion of user annotations with visual content to lead to much improved geo-location inference.

HEATMAPS:
The Google Maps Android API Utility Library[5] includes a heatmap utility, which you can use to add one or more heatmaps to a Google map in your application. Heatmaps make it easy for viewers to understand the distribution and relative intensity of data points on a map. Rather than placing a marker at each location, heatmaps use color to represent the distribution of the data.

3. PROPOSED SYSTEM

The proposed system enables users to report any issues in their locality by capturing images and uploading it to the system. The images contain geo-tagged information, essentially the GPS coordinates embedded in EXIF meta-data within image files which will be used to pinpoint the reported issue on an interactive map. The system consists of both web and android platform which can be used to view and report issues respectively. The proposed system is designed to be modular and draws inspiration from various existing systems. It extends several features of existing systems to insure a user-friendly experience built on top of widely available and existing technologies to incur no extra costs.

The proposed system consists of various modular components such as a user system to enable sign-up, sign-in and maintain minimal profile information of users. Users can view all the data collected over a period of time and also track the status of their reported instances in real time. An interactive map is published where one can view reports in real-time with dynamic report visibility based on social engagement, time of report and trust factor of reporter. Most of the data is stored in a NoSQL data store such as MongoDB which coupled with a NodeJS API server, forms the core of proposed system.

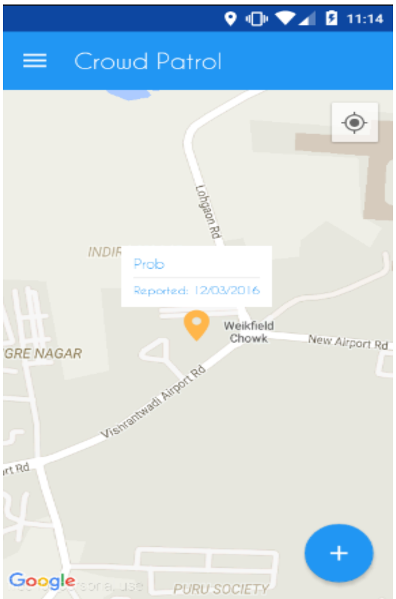

Figure 1: Proposed map interface

The system also consits of an administration module for moderation of new reports, banning abusive users, generating reports etc. Overtime large amounts of data(reports) are collected and analyzed; then visualized for easier consumption and published to the web where anyone can access it. The system also aims to integrate different social features such as – upvote, share etc to promote participation and increase visibility of reports.

Database models are normalized to encourage logically grouping, reduce data redundancy, ensure atomicity and integrity of transactions.

Table 1 : Database Models

Database Model	Description
User	Minimal user profile information, access token, password hash
Marker	Latitude, longitude, timestamp, reporter uid, upvotes, flags, dynamic visibility score
MediaFile	Image path, image resolution, EXIF data, associated report id

4. SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure,behavior and projects different views of a system design. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors.

In the center of system architecture lies the core of the system as shown in figure 1. It is composed of a NodeJS API server that serves data via REST API over https requests. The data is stored in a NoSQL data store such as MongoDB that is connected to API server over standard TCP/IP socket connection. The architecture is easily scalable and can work well in a distributed environment. The API server contains most of the business logic and access control logic.

Users may sign-in via standard email, password combination or opt for social login via Facebook, Twitter. The Map module uses Google Maps API to display an interactive map with different user reported issues within a certain pre-defined radius. The server computes visibility of reports based on trust factor of reporter, timestamp of the report, GPS coordinations contained within EXIF data of image etc.

The visualization module converts data points obtained from data analysis into graphical representation such as bar-plots and heat-maps for easier consumption which can then be exported to desirable format such as CSV, excel sheets.

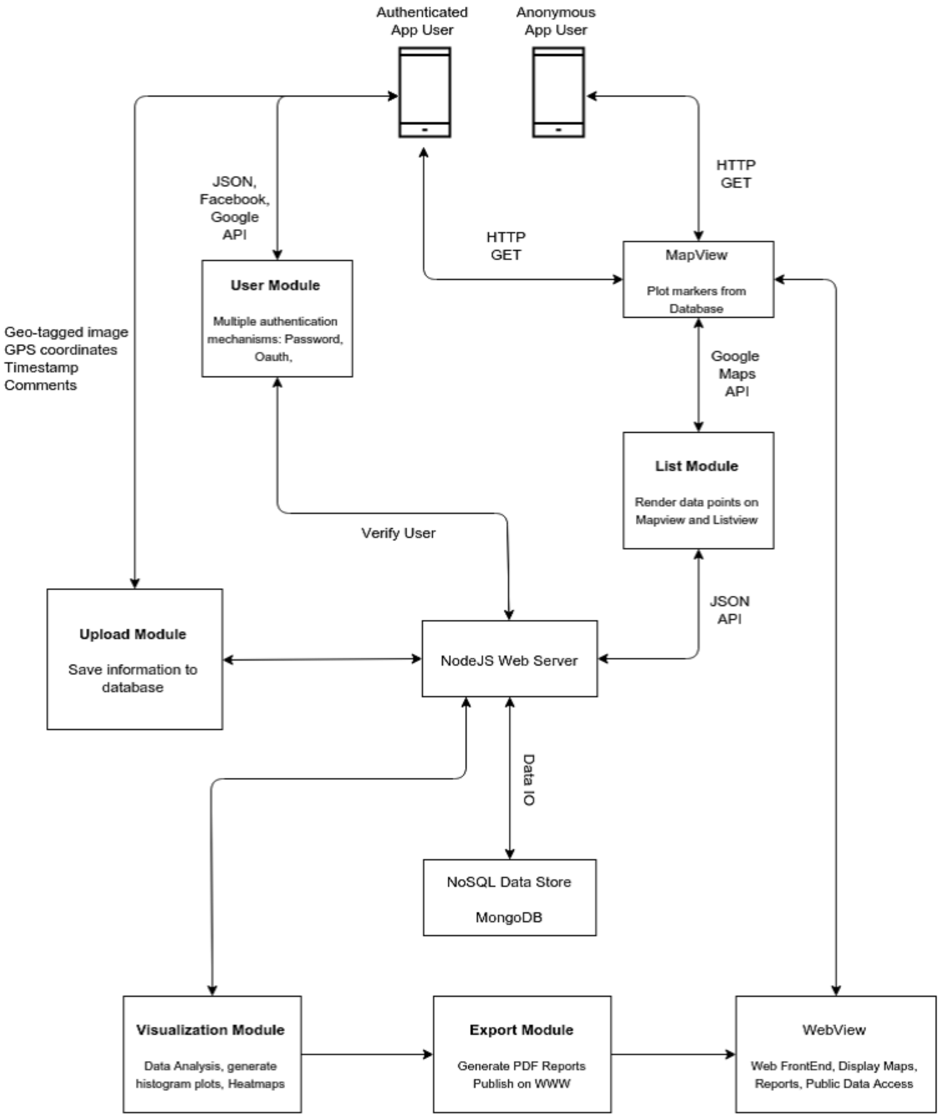

Figure 2: System Architecture of Crowd Patrol

5. ALGORITHMS

The following algorithm, inspired by Reddit listing algorithm[7], is used to compute visibility of a report in order to determine the position and order of reports as they appear on the interactive map.[7]:

```
epoch := timestamp('1/1/1970')
```

```
function diff(date) {
  d = date - epoch

returnd.days * 86400 + d.seconds + d.microseconds)/1000000

}

// Compute visibility of a report

// up : number of upvotes to a reported
// flag : number of times the report has been flagged
// date : date-time when report was submitted
// return : a visibility score of reported
function visibility(up, flag, date) {
score = up - flag
order = log10(max(abs(score), 1))
if ( s > 0 ) {
sign = 1
  } else if ( s< 0 ) {
sign = -1
  } else {
sign = 0;
  }
secs = diff(date) - 1134028003
return round(sign * order + secs / 45000, 7)
}
```

Algorithm for kernel generation (for weighted data points) when generating heatmap overlay on top of Google maps:

```
functiongenerateKernel[] ( radius, sd) {
kernel[] = {radius * 2 + 1}
for (i = -radius i <= radius ++i) {
kernel[i + radius] = e ^ ((-i * i) / (2.0 * sd * sd))
    }
return kernel
}
function convolve[][] ( grid[][], kernel[][] ) {
radius = floor( kernel.length / 2 )
dimOld = grid.length
dim = dimOld - 2 * radius
lowerLimit = radius
upperLimit = radius + dim - 1
intermediate[][] = {{dimOld, dimOld}}
    x = 0, y = 0, initial = 0, val = 0
for (x = 0 x <dimOld ++x) {
for (y = 0 y <dimOld ++y) {
val = grid[x][y]
if not val  == 0.0 {
xUpperLimit = (upperLimit< x + radius?upperLimit:x + radius) + 1
initial = lowerLimit> x - radius?lowerLimit:x - radius
for (x2 = initial x2 <xUpperLimit ++x2) {
intermediate[x2][y] += val * kernel[x2 - (x – radius)]
      }
          }
        }
    }
outputGrid[][] = {{dim, dim}}
for(x = lowerLimit x <upperLimit + 1 ++x) {
for(y = 0 y <dimOld ++y) {
val = intermediate[x][y]
if not val == 0.0  {
yUpperLimit = (upperLimit< y + radius?upperLimit:y + radius) + 1
initial = lowerLimit> y - radius?lowerLimit:y - radius
for (y2 = initial y2 <yUpperLimit ++y2) {
outputGrid[x - radius][y2 - radius] += val * kernel[y2 - (y – radius)]
              }
          }
      }
    }
returnoutputGrid
```

6. CONCLUSION AND FUTURE SCOPE

In conclusion, the proposed system turns out to be more cost effective by utilizing the power of crowd to report different social issues while making use of existing technology ecosystem. From an administrative point of view, a widely available technology stack and seemingly easier implementation and deployment means most civic bodies can seamlessly adopt, deploy and maintain the system.

During testing of initial prototype, it was observed that the simple design flows and user-friendliness of the app resulted in quicker and wider adoption by both technically adept users and laymen.

The information on market share from statista.com as depicted in figure 3 shows sufficient market penetration by Android smartphones at about 45% of total mobile operating systems which further strengthens our conclusion of wider adoption of the platform.

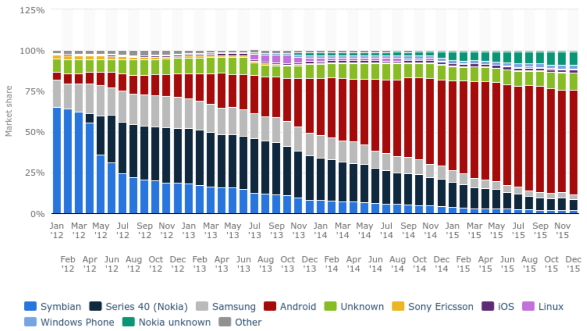

Figure 3: Android market share 2015

The transparent nature of the system, lacking in existing solutions, helps reduce administrative costs and also instils trust in users.

Furthermore, the system can be extended to include a range of other issues that require crowd participation such as reporting instances of disease, accident-prone areas, civic issues, tagging stray animals et cetera.

7. ACKNOWLEDGEMENTS

This work is affliated by the Department of Computer Engineering, G. H. Raisoni College of Engineering and Management (GHRCEM) – Pune, Maharashtra. India. We would like to thank our project guide Mrs.SeemaUdgirkar, HOD Mrs.Poonam Gupta and all faculty of Computer department, GHRCEM for their valauble guidance and help.

8. REFERENCES

[1] Kubásek, M. (2013). ―Mapping of Illegal Dumps in the Czech Republic-Using a Crowd-Sourcing Approach.‖, vol., no., 30 May. 2013

[2] Sen, R., "RasteyRishtey: A social incentive system to crowdsource road traffic information in developing regions," in Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on , vol., no., pp.171-176, 6-8 Jan. 2014

[3] Heipke, Christian. "Crowdsourcing geospatial data." ISPRS Journal of Photogrammetry and Remote Sensing 65.6 (2010): 550-557

[4] Suri, Manik V. "From Crowdsourcing Potholes to Community Policing: Applying Interoperability Theory to Analyze the Expansion of ―Open311." Berkman Center for Internet & Society at Harvard University (2013).

[5] Google Maps API intro http://developers.google.com/maps/documentation/android-api/intro

[6] J. Prasanna Kumar "Inferring Location from Geotagged Photos" - IJARCSSE, ISSN: 2277 128X - Vol 4, 9 Sept. 2014.

[7] https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9