Crowd and Event Detection by Fusion of Camera Images and Micro Blogs

Sohei Kojima　Akira Uchiyama　Masumi Shirakawa　Akihito Hiromori　Hirozumi Yamaguchi　Teruo Higashino
Graduate School of Information Science and Technology, Osaka University
Suita, Osaka 5650871, Japan

Abstract—In this paper, we propose a new application to infer the "cause" of human crowd (scheduled events, sudden accidents and so on) by mobile crowd sensing. The idea is that we leverage our phone-camera-based, crowd-sourced people counting to firstly localize an event with human crowds, and then extract keywords that spatiotemporally correspond to the event from micro blogs such as tweets. Such keywords are further analyzed to find out the most-frequent ones, which can be used to characterize the detected human-crowd event and to estimate its cause. We demonstrate our prototype design using real camera images and tweets to automatically detect the Halloween street party in Tokyo and estimate its human density.

I. INTRODUCTION

In urban areas, understanding event occurrence with their scale is important for various smart city applications such as transport and logistics optimization, accident prevention, urban facility planning and marketing. With the widespread use of social media such as Twitter and foursquare, the mainstream of event detection is to leverage them [1], [2]. In such approaches that focus on data analysis in cyberspace not only event occurrence but also event types (e.g. accident, sport game, and party) can be estimated by analyzing phrases directly input by users. However, such SNS-based approaches have certain limitations for physical event detection. Firstly, the event identification and localization by SNS analysis is not straightforward as user posts like tweets and images do not usually contain sufficient information to identify event locations, scales and other physical characteristics such as human density. This is very natural since they are not intended to accurately point out the event, but intended to tell friends and others the occurrence and contents of the events. This can be proved by the fact that quite few tweets have geolocation in reality. Secondly, the number of samples are often limited to precisely recognize event locations and estimate physical characteristics [3].

On the other hand, crowd monitoring by cameras is more promising directly estimate event occurrence in spot-level or street-level. In March 11, 2011 in Japan, after the Great East Japan Earthquake, stations and streets were occupied by full of people who could not go home by public transportation (almost all trains were out of service). Such a mass of people obstructed vehicle traffic and finally Tokyo downtown was paralyzed. Lessons learned from this case are that crowd and congestion information should be obtained and provided to those people to assist their right decision making. Similar cases may also happen not only in such extraordinary situations, but in our daily life. When a huge event is held, people crowd information would be of great help in planning optimal logistics. In order to address the limitations of CCTV coverage in cities, we have proposed a crowd sensing method [4] using a smartphone camera to estimate the number of people in a scene captured by a single user. In this crowdsourcing approach, we ask some cooperative users to take photos and the app on their smartphones can automatically detect people in the photo shots and estimate the number of people. This information is reported to a cloud server with geolocation information. Consequently, this system focuses on the physical aspect of crowd-related events with their pinpoint locations. Taking photos do not overload the cooperative users, and in particular, one photo shot can cover a wide area. Therefore, this scheme can work with the limited number of cooperative users, but we do not want to request those users to input texts describing the situations, which imposes additional efforts. We believe such context information should be obtained by SNS spontaneously posted by a number of users, by proper filtering of unrelated ones. If this fusion of crowd-based event detection in the physical world and SNS-based context filtering in the cyber space is realized, the crowdsourcing of human event detection becomes much more powerful to estimate the real world event occurrence.

Therefore, in this paper, we propose a new application to infer the "cause" of human crowd (scheduled events, sudden accidents and so on) by mobile crowd sensing. The idea is that we leverage our phone-camera-based, crowd-sourced people counting [4] to firstly localize an event with human crowds, and then extract keywords that spatiotemporally correspond to the event from micro blogs such as tweets [5]. In order to detect events from twitter, we use TF-IDF (Term Frequency-Inverse Document Frequency) which is often used in text mining as an indicator of trend. The estimated counts and the TF-IDF scores have similar trends, so the highest TF-IDF score keyword can be used to characterize the detected human-crowd event and to estimate its cause. By this method, crowd information from the camera image of the smartphone which is a physical system is fused with event information existing in the cyber system. We describe our prototype design and demonstrate an example use case using real camera images and tweets to detect the Halloween street party in Tokyo. The results show that our approach is promising to capture scale.
and types of events by a crowd sensing approach.

II. RELATED WORK

A. Infrastructure-based People Counting

There have been several approaches to crowd sensing using various sensors. For example, there are vision-based pedestrian tracking approaches using CCTV cameras [6], [7], passive infrared (IR) sensors approaches [8], [9], laser range scanners (often called LIDARs) approaches [10] and so on. However, they have a severe limitation on coverage of monitoring since they need pre-installed sensing infrastructure. In addition, IR sensors and LIDARs approaches may not perform well in heavily-crowded situations, where sight of the these sensors are often obstructed by pedestrians near the sensors.

B. Mobile-based People Counting

A more cost-efficient approach to crowd density estimation is to utilize sensors in commercial mobile devices (e.g., smartphones). Refs. [11], [12] and [13] introduce an approach to estimate and visualize human mobility from cellular network traffic. However, these approaches target city-level crowd and human detection. Kannan et al. [14] count the number of mobile phone users in a crowd by exchanging audio beacons between phones in proximity. The audio beacons contain frequency components that are associated with the phone’s own ID (e.g., MAC address) and the IDs of detected neighbor phones. By repeating such beacon exchange until the set of detected neighbors converges, the phones can recognize the number of phones in the crowd. Although it can count up to hundreds of phones by carefully designing the coding algorithm for the audio beacons, it can recognize only the mobile phone users who participate in the crowd counting service, keeping their speakers and microphones on. Weppner et al. [15] employ short-range wireless ad-hoc communication via Bluetooth for participatory sensing of crowd density with mobile phones. To mitigate dependence on the proportion of mobile phone users who enable Bluetooth of their phones, they use variance in RSS values as well as the number of detected neighboring phones to classify the current crowd density in a target area into 7 categories. However, accuracy of crowd density estimation still significantly depends on the ratio of Bluetooth-enabled phones.

C. Our Motivation

In addition to various people counting approaches as we discussed so far, there are many effective approaches to understand characteristics of geographic regions and specific locations based on crowd sensing. ConvenienceProbe[17] analyzes trajectory data offered by mobile phone users to identify retail trade areas. Chon et al.[18] designed a framework to automatically characterize places based on opportunistically-captured images and audio clips from smartphones. These techniques are useful to recognize situations in terms of various contexts. TweetMotif [1] groups messages by frequent significant terms which facilitate navigation and drilldown through a faceted search interface. The topic extraction system is based on syntactic filtering, language modeling, near-duplicate detection, and set cover heuristics. iSee [19] which is not Twitter based event detection method detect and localizes specific events by participatory sensing with mobile phones.

In this paper, we design a concept application which fuses multiple crowd sensing methods: camera-based people counting using smartphones and text data mining from Twitter. Our motivation is to show the effectiveness of information fusion in mobile crowd sensing. We also investigate limitations and challenges in our concept design through a case study using real images and tweets collected during the Halloween street party in Shibuya, Tokyo.

III. SYSTEM DESIGN

A. Overview

Figure 1 shows the overview of our system. A user takes a picture of a people crowd (i.e. an event) by our application from a height. The number of people in the crowd is estimated and sent to a server with its location information obtained by manual input or location services such as GPS and WiFi. Our application dose not upload images, so there is no privacy concern. To recognize the event type, the server collects tweets with the corresponding location information. Since most of tweets are without geo tags, we can use location names included in tweets instead. To do so, we obtain a location name from location information provided with the people count by using Google Maps API which converts a pair of a latitude and a longitude into the corresponding address or landmark names nearby. Then, we use some of the landmark names and/or a part of the address as the location names and extract tweets with them, assuming some of the extracted tweets are related to the event captured by the user’s smartphone.

We assume an event is categorized into either scheduled or unscheduled events. Examples of scheduled events are festivals, elections and concerts while those of the unscheduled events accidents which happen unpredictably. Since noise
filtering is important in tweet analysis, we prepare a database of event keywords of the both types for extracting event-related tweets. In this way, our system can estimate the both event types as long as they are captured by users and spatially and temporally unique. Finally, the detected events are shown onto a map for visualization shared with other users.

B. People Counting

In our people counting method [4], users take two time-consecutive images of crowds at a short interval by smartphones in bird’s-eye view from slightly high places such as footbridges and the second or higher floors of buildings as shown in Fig.2. Since conventional human body detection from images does not often work in crowd detection, we take a simple approach where the number of people is estimated by a pixel-to-people function which is built based on the results of extensive simulations on the Unity human model [20]. In order to identify people presence areas in images, “moving object region” (MOR) is obtained by the well-known background subtraction method from two images taken consecutively as shown Fig. 3. Figure 4 depicts an example of MORs overlaid with one of the two consecutive images.

In ref [4], we have shown that the average estimation error in ideal simulation cases is 12.3% while it is 25.3% in the experiments using real images taken by volunteers with heights and angles estimated by smartphone sensors. We have revealed both advantages and limitations of our method in terms of accuracy of people counting. Then, we have also shown that our estimation is applicable to flow density estimation in Osaka downtown with only 13.8% error. In this paper, we fuse crowd density and location estimated by our people counting method with event information obtained by Twitter in order to detect events with their types and scale.

C. Event Detection

We utilize Twitter as a source of event types. Our system employs an event detection method proposed in Ref. [5] to extract trend keywords every one hour. In this section, we briefly explain the key idea for event detection.

An event is regarded as a trend limited in space and time. This means we need to detect trends for event detection. We use TF-IDF (Term Frequency-Inverse Document Frequency) as an indicator of trend. TF-IDF is often used in text mining to measure a weight of a term \( t \) in a document set \( D \) and defined as below:

\[
\text{tfidf}(t, d, D) = \frac{\text{tf}(t, d)}{\text{idf}(t, d, D)} = \frac{n_{t,d}}{\sum_{d \in D} n_d} \cdot \log_2 \left( \frac{|D|}{|\{d : d \ni t\}|} \right)
\]

\( n_{t,d} \) is the number of appearances of \( t \) and \( n_d \) is the number of words in a document \( d \). Then, trend keywords are extracted as a set of top-N keywords in TF-IDF.

We can detect events in an area \( a \) by applying trend detection for a set \( D_a \) of tweets related to \( a \). For extraction of \( D_a \), as we mentioned in III-A, we use Google Maps API to obtain the address of \( a \) and landmark names around \( a \) from location information provided by a user.

In this paper, we simply regard the top trend keyword as the one which represents the cause of the event detected by our people counting method.

IV. CASE STUDY

We have evaluated our system by using images captured from a live streaming video of Shibuya, which is one of the major area in Tokyo. The video is provided by Shibuya Television with the permission of the company [21]. An example of the capture image is shown in Fig. 5. We captured images at 2PM and 8PM on October 30 and 0AM, 1AM and 8PM on October 31. In this evaluation, we focus on a Halloween street party in Shibuya which is a scheduled event. According to the company, we set the height and the tilt angle
of the camera to 9 [m] and $\frac{\pi}{18}$ [rad] for our people counting method.

**A. Prototype Implementation and Example Use Case**

We have implemented our android application by OpenCV which visualizes detected events and their scale on a map. A snapshot of the application is shown in Fig. 6. The event scale is visualized by a heat map and helpful for users. In addition to the estimation result of the number of people, the detection result of the event can also be shown on the map. By this application, users mark their location and target location in which crowd is. After marking, users take two time-consecutive images of crowds at a short interval by smartphones in bird’s-eye view. Our application automatically compute the number of crowd from two images.

**B. People Counting Result**

From this section, though not our application’s original usage, we use images captured from a live streaming video of Shibuya to evaluate the people counting and the event detection. To see the performance of people counting, we show the change of the estimated number of people over time in Fig. 7. We note that the area captured by the camera is a large intersection and therefore there are two time periods when pedestrians cross the intersection and wait for a signal.
TABLE I: Example of Estimation Result

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation Result</td>
<td>105.8</td>
<td>99.8</td>
<td>91.1</td>
<td>79.9</td>
<td>114.2</td>
</tr>
<tr>
<td>Ground Truth</td>
<td>155</td>
<td>176</td>
<td>112</td>
<td>93</td>
<td>223</td>
</tr>
</tbody>
</table>

Fig. 8: High Occlusion Example

to change. Since our method assumes people in a crowd are moving while taking pictures, we can see clear oscillations of the estimated people counts due to the cycle of the traffic light.

Since deriving the actual number of people in an image requires much human effort, we chose the highest peak of each time in Figs. 7(a)-(e) and manually counted the actual number of people in the images. Table I shows the ground truth and the estimated number of people.

From Table I, we see the errors in the cases (b) and (e) are particularly large. To investigate the reason for this, Fig. 8 shows the image used for the estimation of the case (e). By carefully analyzing the image, we have found that traffic was controlled by police and the areas open for pedestrians are limited. Therefore, the areas were extremely dense which is over the limitation of our people counting due to frequent occlusion (i.e. overlap of people in images). However, we can still understand some less-dense cases such as (d) and the fact that the area is very crowded from our estimation results.

C. Event Detection Result

To see the performance of event detection from Twitter, we collected tweets with “Shibuya”, the address of our target area, from 9AM on October 30 to 9AM on November 1. We also collected tweets with “Shibuya” on other days for event detection. Table II shows TF-IDF scores of the top-2 keywords related to events. The result indicates that one event is sometimes related to multiple terms (e.g. Halloween and “dress up”). Therefore, we may use some ideas such as relationships between events and terms for more accurate event detection. Figure 9 shows the TF-IDF score transition of “Halloween” and the number of tweets from 9AM on October 30 to 8AM on October 31. TF-IDF scores and the number of tweets have similar trends in Fig. 9, so we can detect the time of the crowd occurred by only the number of tweets.

We also compared the estimated people counts and TF-IDF scores of “Halloween”. We regard the average of all peaks in each case (a)-(e) as the estimated count. For comparison, we normalized TF-IDF scores so that the normalized TF-IDF score of (e) becomes equal to the estimated count of (e). The result is illustrated in Fig. 10.

From the result, we see that the estimated counts and the TF-IDF scores have similar trends in the cases of (b)-(e). However, the estimated count is very different from the TF-IDF score in (a). This is because (a) is the image captured at 2PM on Sunday when there are many people who are not related to the Halloween event. Nevertheless, our system can successfully detect the Halloween street party event in Shibuya along with its scale.

V. DISCUSSION

In section IV, we demonstrated that we can detect the scale of people crowd in Shibuya and it is due to Halloween by combining our people counting method and text mining from social media. However, a keyword corresponding to an actual event captured by our application does not always have the highest TF-IDF score, which is one of our limitations. Nevertheless, fusion of multiple crowd sensing methods improve the quality of information by filtering out unrelated keywords of which TF-IDF scores are low.

Another issue is an algorithm design to choose appropriate name(s) for text mining from Twitter. In our concept, as we described in section III-A, we use Google Maps API to obtain an address of a target area when we collect tweets related to the target area. However, in our case study in section IV, we directly used the name of the district (i.e. “Shibuya”) for Twitter search. It is obvious that the location name “Shibuya” is included in the result of the Google Maps API query. However, “Shibuya” is only a part of the whole address which includes names of a city, a street, a prefecture and

TABLE II: TF-IDF Score

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halloween</td>
<td>3640.9</td>
<td>8713.49</td>
<td>8969.9</td>
<td>4840</td>
<td>11303.5</td>
</tr>
<tr>
<td>dress up</td>
<td>1769.6</td>
<td>3316.8</td>
<td>3126.0</td>
<td>1578.9</td>
<td>5997.8</td>
</tr>
</tbody>
</table>

Fig. 9: TF-IDF Transition
so on. Therefore, it is still challenging to choose appropriate area name(s) from the whole address. To solve this problem, for example, we may define the names with their locations in advance and choose the nearest name from the location obtained by location services such as GPS and WiFi.

VI. CONCLUSION

In this paper, we proposed a design of a smartphone application to detect events along with its people count by fusing our smartphone-based people counting method with an event detection method from Twitter. We demonstrated our application in the Halloween street party in Shibuya, Tokyo as our example use case. The use case showed that the event was detected successfully by fusing our crowd-based people counting method with another source from crowd, i.e. Twitter. It also indicated that the amount and the quality of information are enhanced by fusion of cyber (e.g. social media) and physical systems (e.g. people counting).

For future works, we are planning to further collect tweets for a longer period of time and evaluate our system for some unscheduled events as well as other types of scheduled events. We also need to evaluate the unscheduled events such as accidents and disasters. In order to detect such events, detecting “bursts” of trends is necessary by collecting tweets continuously and filtering keywords by an unscheduled event database.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI JP26220001, JP15H02690 and JP26700006.

REFERENCES


Fig. 10: Comparison between Estimated Counts and TF-IDF Scores