Filters for Wi-Fi Generated Crowd Movement Data

Cristian Chilipirea, Andreea-Cristina Petre, Ciprian Dobre
Faculty of Automatic Control and Computers
University “Politehnica” of Bucharest
Bucharest, Romania
{cristian.chilipirea; ciprian.dobre}@cs.pub.ro;
andreea.petre@cti.pub.ro

Maarten van Steen
CTIT
University of Twente
Enschede, Netherlands
m.r.vansteen@utwente.nl

***Abstract*—Cities represent large groups of people that share a common infrastructure, common social groups and/or common interests. With the development of new technologies current cities aim to become what is known as smart cities, in which all the small details of these large constructs are controlled to better improve the quality of life of its inhabitants. One of the important gears that powers a city is given by traffic, be it vehicular or pedestrian. As such traffic is closely related to all other activities that take place inside of a city. Understanding traffic is still a difficult process as we have to be able to not only measure it in the sense of how many people are using a particular path but also in analyzing where people are going and when, while still maintaining individual privacy. And all this has to be done at a scale that would cover most if not all individuals in a city.**

With the high increase in smartphones adoption we can reliably assume that a large part of the population in cities are carrying with them, at all times, at least one Wi-Fi enabled device. Because Wi-Fi devices are regularly transmitting signals we can rely on these devices to detect individual’s movements unobtrusively without identifying or tracking any particular individual. Special sensors that monitor Wi-Fi frequencies can be placed around a city to gather data that can later be used to identify patterns in the traffic flows.

We present a set of filters that can be used to minimize the amount of data needed for processing and without negatively impacting the result or the information that can be extracted from this data. Part of the filters we present can be deployed at the sensor level, making the entire system more scalable, while a different part can be executed before data processing thus enabling real time information extraction and a broader temporal and spatial range for data analysis. Some of these filters are particular to Wi-Fi but some of them can be applied to any detection system.

Keywords—Wi-Fi; Crowd Sensing; tracking.

I. Introduction

Smart cities form an important focus of current research. People are envisioning more and better ways in which technology and information can improve our everyday life. To do so, we need to understand that life. In a city, traffic constitutes one of its crucial elements. Understanding traffic flows may help us to improve city life. In this paper, we concentrate on measurements on flows of pedestrians. In particular, we are interested in automatic and unobtrusive measurements on how pedestrians move through a specific designated area.

Tracking groups of people, understanding traffic flows, making sense of population densities, and so on, can be done in different ways. One solution takes advantage of the fact that many individuals carry smartphones or similar devices that are not only Wi-Fi enabled but also transmit Wi-Fi packets at regular intervals (for instance to search for new Wi-Fi hotspots). By deploying sensors that gather these packets we can gain more insight in pedestrian behavior.

Let us consider each Wi-Fi packet received at a sensor as a data point that we can later analyze. The number of data points depends on the number of people that carry Wi-Fi devices (we believe it to be correlated with the total number of people), the amount of Wi-Fi traffic these devices produce (dependent on the owner’s usage of the device) and finally with the number of sensors that are deployed. Cities are large population centers with a high number of individuals, occupying large areas, requiring potentially tens to hundreds of sensors to cover the area of interest. As a consequence, the amount of data that we need to process can easily grow dramatically. Moreover, when having to deal with very large data sets, real-time analysis can become impossible, excluding many interesting applications, such as, for example, real-time crowd control by guiding people through less crowded areas.

In these detection systems there are a number of sensors placed around an area of interest. The sensors gather all Wi-Fi packets, construct detections and forward this data to a centralized server. The server then processes the data and creates visualization, analysis and/or long term storage. This architecture is common over all similar systems we encountered in this type of projects.

We can expect that the number of Wi-Fi devices will continue to grow, this is a clear trend given by the smartphone industry and by the increased interest in Internet of Things. We can also expect the number of sensors to grow. For instance whereas most projects still deal with few sensors (3-5) and small areas such as botanical gardens or beaches [1], public events such as concerts [2], campuses [3] grow to large scales of hundreds of sensors and billions of detections like in the case of [4] where the monitored area spreads over a large hospital complex. This trend will continue and reach metropolitan scales like is the case with other methods such as using Call Detail Records [5], where using Wi-Fi could give better resolution and accuracy in movement tracking. Reducing the size of sets of raw Wi-Fi device detections is essential.

In this paper we present a number of filters and show how these filters can drastically reduce the size of Wi-Fi-based device detections. We show how these filters work, what their purpose is and what effect they have on data gathered at two deployments of these systems. Some of the filters can be deployed at sensor level, lowering the bandwidth usage to gather data from multiple sensors and permitting a better scalability with regards to the number of sensors. Others can be applied only on large sets of data but have a large impact on the size of the data set. With the reduced data size processing of this data can be executed at higher speeds, maybe even enabling real-time, large-scale processing.

For some of the filters we present here we show not only their usefulness in reducing the data set but in cleaning noise out of it, improving its quality.

II. Related Work

Understanding crowds and their movement has been an interesting research topic for some time. The benefits it can provide in transportation, simulations, and improving day-to-day activities has kept it as a main focus for research. Popular methods for understanding crowds are using visual analysis of camera feeds. An example of such a system can be seen in [6]. A similar system that uses cameras to measure the number of people in an area and to model small crowds through merging and splitting events is [7]. These systems can work at different scales. Previously we gave examples of systems that work at the size of crowds, but there are also systems that treat individuals movements such as described in [8]. An overview of visual systems can be found in [9]. We mention that visual systems also require filters for their data like in the case of [10] where the crowd is filtered out to reveal object left behind such as bags or packets.

The results of camera vision systems that track individuals can be used to detect human behavior. One solution for this uses models [11]. Extracting models of human behavior is one important output of crowd monitoring. These models have many uses in games and entertainment or medical and architectural aplications as stated in [12]. Many try to create better models like in the case of [13] or [14]. But without real-life measurements these models still lack realism. In [15] models are created that manage to mimic real-life measurements and offer a more realistic results. However these models use the scale of the entire city, here the model makes the correct assumption that humans have favorite locations where they spend most time, like home or work. After the models are created and refined enough they can be used in all kind of simulations such as to better identify opportunistic network algorithms [16].

When scale is required, video streams are not a valid option. Because of this many projects are researching other methods of extracting movement data without the use of camera systems. One example [17] uses data from a device carried around by individuals that gathers Wi-Fi and GPS signals. This data gives insight in human mobility and features of a city such as Access Point popularity in different areas. However, when there is a need to gather data from even more people using a device or an application installed on individual's phones is not acceptable. Because of this Wi-Fi systems have been built that manage to track humans without the need for them to be part of the system. Some works use patterns in signals on the Wi-Fi frequencies to identify individuals walking [18], groups [19] or even to count how many people are part of a group [20].

Different works focus on using Wi-Fi packet detectors, these are hotspots that act as sensors listening to packets in accordance to the 802.11(a/b/g/n) standard. Most Wi-Fi packets contain a MAC address permitting tracking of a device over multiple sensors. The advantage here is that most people already own a smartphone capable of communicating via Wi-Fi and they carry these devices with them most of the time. The disadvantage is that only people that have Wi-Fi enabled on their mobile device can be tracked. These systems have high popularity for indoor environments as can be seen in [21], [22] and [23] where not only localization is achieved but it is done with a high degree of accuracy, with errors less than 1 m. These systems can even be used to measure queues of people and their dynamics [24].

The systems that we are most interested in use Wi-Fi packet detection to measure movements over large areas (more than a few buildings). A good example is given by taking such measurements a botanical garden or a beach [1].

In our research we noticed that the authors of such works, to improve quality of sensed data, often use data filters as at least one step in their processing of the data. In [2] devices that do not appear at multiple hotspots are filtered out because they cannot show movement information if added to the final data set. This is one of the filters we will present in more detail in this paper. Duplicate detections (detections at multiple hotspots at the same time) and static devices (Wi-Fi enabled printers) are filtered out in the work presented in [3] where the data is used to analyze movement inside a campus. Finally [4] filter out devices that are known to be part of the buildings infrastructure and staff; they also filter indoor detections from outdoor ones.

The location of filters in the processing stream is discussed in detail in [25], where they show how distributing the computation of the filters can dramatically improve the processing time of applications. This is similar to our solution of filtering data as early as possible, in our case on the sensors that detect the Wi-Fi packets that we will describe in the next section.

III. Filtering at the sensor

We deployed a small set of Wi-Fi packets sensors in the cities of Arnhem and Amstelveen (The Netherlands). These sensors are hotspots that monitor Wi-Fi and log all detected packets (meaning mostly Wi-Fi protocol headers; for security implications, we do not look inside the communication taking place). In our architecture, all logged packets are further sent to a centralized server, where further filtering discussed in the next section, and data processing algorithms are used (i.e., for tracking crowd mobility and crowd control applications).

When trying to achieve scalability in our system our main concern was the bandwidth utilization between the hotspots and the central server that gathers the data. To minimize bandwidth usage and control to some degree the quality of sensed data, we implemented three filters that would minimize the amount of packets we consider to be detections. These filters also have a correctness role: for instance, we do not want to consider devices that are not mobile, like a laptop that is permanently in use and in range of one of our hotspots, or a different hotspot.

- filter 1 - this filter accepts packets that have a *transmitter MAC address* and that MAC address is of a wireless device. Not all packets have a transmitter address and without one we cannot know who we assign the detection to; without it, we cannot track the device along multiple detections or multiple hotspots. We also mention that we are only interested in wireless devices, this is especially true in the case of data packets, and data packets have two bit fields named “from DS and to DS". These fields indicate if the packet is coming or going towards the wired distribution system. If the field "from DS" is set to 1 then the packet is coming from a device that is in a wired network with the access point used in the wireless communication. We are not interested in these wired devices and we filter them off. We believe that in other works [26] this filter might be missing and is causing the appearance of "Mystery OUIs". This filter is a correctness one, as it does however also make the entire system use less bandwidth and resources. The filter is also extremely fast: it only needs to check the type of the packet and in case it is a Data packet it needs to check the "from DS" field and this is all that is needed to make the decision if a packet passes the filter or not.

- filter 2 and 0 - Filter 2, also a correctness filter, eliminates all packets that come from access points. It is important to have the filter because in our case studies we encountered packets that have "from DS" set to 0 and the packets themselves have an access point as a transmitter. We know it was an access point because we encountered "Beacon" packets with the same MAC Address. Filter 2 does need to have a list with all encountered access points and this list is generated by filter 0. Complementary, filter 0 makes a list with all the MAC addresses of the "Beacons" it received, these packets are only sent by the access point and have their address in them, filter 0 also eliminates all "Beacon" packets. We mention here that all packets that are eliminated by filter 0 would have also been eliminated by filter 1 because we know "beacon" packets have no transmitter address different than the BSSID. The list of "Beacon" transmitter addresses saved by Filter 0 has a maximum size of 50. We chose 50 because we wanted it to be small and we do not expect that many sensors in range of one of our hotspots.

- filter 3 - this is not a correctness filter but it does offer high efficiency gains. This filter is temporal as it filters all the packets that have the same transmitter address as a detection made in the last 3 seconds. The 3 second interval was chosen empirically. For instance in the work of [26] a 1-second interval is used as an aggregation point that has the same effect as our filter. The larger the time frame is, less packets will be detected and less data will be sent to the central server; however this comes with a loss in accuracy. Another part of this filter is focused on correctness and it filters all the devices that we have seen for more than a few hours, devices that we consider non-mobile. We chose the number of hours to be 5, but any reasonable amount can be used here. To be able to account for all detected devices this filter keeps a log with all the MAC addresses of the transmitter of all the packets that are considered detections. This log is kept in the form of a hash table with a statically allocated size.

We found empirically that filters work best in 0, 1, 2, and 3 ordering, this is also forced by some of the dependencies they have on each other. All the packets that pass all three filters are considered to be a detection. For these packets the transmitter’s MAC are sent to the centralized server.

To evaluate our filters we used two distinct data sets. One data set is obtained on one of the hotspots inside a room of a student complex of VU University Amsterdam. The other data set is obtained in the city of Arnhem from the barXO hotspot. These two datasets are rather different, but they do have a similar number of total packets and size. We obtained the data sets by making a tcpdump on the hotspots that run for a few days. Because the two traces were run in different environments, particularities of these environments can be seen in the data. For instance the student-complex trace has a lots of data packets that are sent by only two devices. We also mention the channels: the student-complex data was gathered on a channel where only one active Wi-Fi network existed, while the Arnhem trace was run on a channel where no devices were actively communicating. The actual number of packets and size of the files as well as the dates on which the trace started can be seen in **Table 1**.

Table 1 Dataset characteristics

	Arnhem	Student Complex
# of packets	2414203	2906574
MB	257	214
Start Date	2014-07-07	2014-07-04

To test our filters we used both data sets but we did not put any time constraint limitations. In both cases we left the software process the data as fast as possible. This means that we analyzed a few days of data in under 3 minutes, and this has some dramatic effects on the effectiveness of the 3rd filter. The 3rd filter is however the only one affected by time, the others are independent.

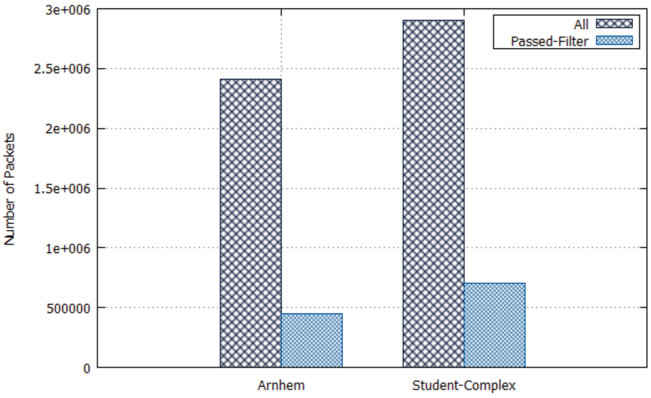

Fig. 1 Filter 1

To have an accurate representation of how the filters functioned we stopped all the other filters and tested them independently. At the end all filters were operational and we tested them as a whole.The first filter eliminates all the packets that do not have a transmitter. As one can see in Fig. *1* in both scenarios the filters managed to reduce the number of packets from over 2.5 Million to about 0.5 Million. To validate the functionality, we also sampled the data and manually inspected the filtered packets, as well as the packets that passed; no false negatives were found in the process. The more effective the filter, the smaller the red bar would be in comparison to the blue one. This filter serves in first place a correctness function, as all the packets that do not have a transmitter cannot be used as a detection. However this is an extremely fast and powerful filter as it eliminates about 80% of the packets.

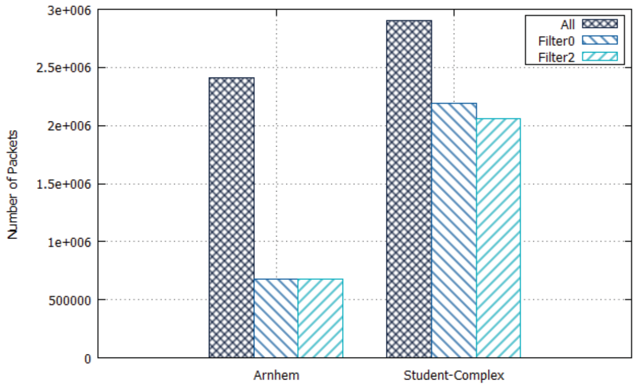

Fig. 2 Filter 0 and 2

The results for filters 2 and 0 can be seen in Fig. *2*. The results vary so dramatically between the 2 scenarios because of the extremely large number of Beacon packets that dominate the Arnhem trace. Filter 2 is a correctness filter. This means that even though the number of packets filtered out by filter 2 in comparison to filter 0 and filter 1 is just minimal, the filter itself should still exist to eliminate all possible detections of non-mobile devices, of access points. Fig. *2* shows how filter 2 works best in an environment where a lot of data traffic is expected. For instance it might prove to be extremely useful in a residential area where people use Wi-Fi to stream movies or other high bandwidth usage content. However in an area with coffee shops where most people just check their e-mail or do small amount of browsing, the filter might not be so efficient.

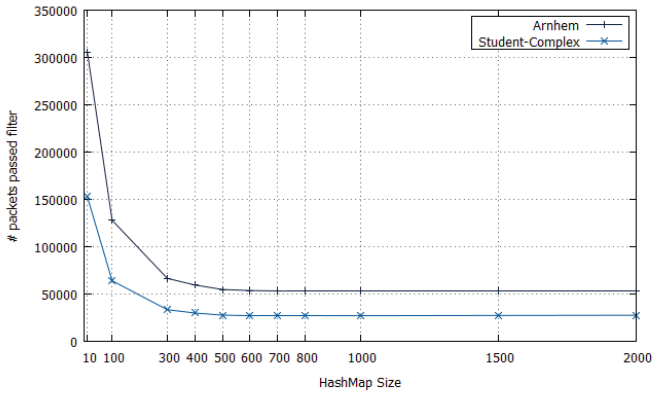

Fig. 3 Filter 3

The final filter eliminates all packets that have transmitter MAC addresses that have been detected in the last 3 seconds. To be able to filter these packets, the filter needs to keep a log of all packets that have been registered as a detection. This filter is kept as a statically allocated hash table, with a maximum length. The size of the hash table directly affects the key calculation and the number of collisions it would have.

To properly test the effectiveness of the third filter we compared the results on both data sets with varying maximum size of the hash table. We do this because we want to use the least amount of memory as possible while still having a maximally effective filter. The results are displayed in Fig. *3*. Here we can see a similar trend for both traces. Because we processed a few days of data in under two minutes and there were not an extremely high number of unique devices in the trace, this filter was very effective, there were a lot of packets with the same transmitter address that were processed in under three seconds, forcing most of them to be filtered.

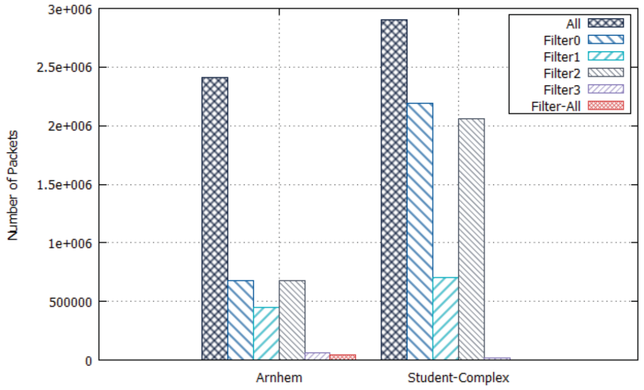

Fig. 4 All Filters

In Fig. *4* we compare all filters. Filter 0 to 3 are started one at a time (note here filter 2 cannot exist without filter 0). Then the last one is with all the filters started at the same time. Here we can easily see that the filters that make most of the difference are filter 1 and 3. Having all filters active minimizes the number of accepted packets even further, this happens mostly because packets that are accepted by filter 1 are not accepted by filter 3, the same is also true in reverse. This proves that all filters are important and that working together a very large number of packets are removed and only the most relevant ones are kept and sent to the central server.

IV. Filtering data after centralization

The form of the data set that we produce is:

sensorid – deviceid – timestamp

Here *sensorid* and *deviceid* identify the sensor (as a way to extract its physical location) and, correspondingly, the detected mobile device (usually in the form of an MD5 hash). We have found that this format is similar across most projects that gather this type of data.

- Filtering duplicates. This filter eliminates all data duplicates. We consider duplicates any two data points that have all three values (sensorid, deviceid and timestamp) equal. Without making a comparison with our approach, we note that other works consider duplicates any two data points with the same deviceid and timestamp (allowing different sensorids).

- Filtering by time. Usually only a part of the data set is of interest to data analysis or the data set needs to be processed in chunks that span over one day or one week. This filter eliminates all data points that have a timestamp that does not fit between two given values.
- Filtering Apple products. As advertised in [27] Apple products randomize their MAC address when sending probe requests to identify new hotspots (These probe requests are the packets that we capture when the device is not connected to a network, and we expect this to be the general case). Because this address is randomized a device using this feature cannot be tracked over multiple sensors. Even worse, two different devices can send out the same MAC address making the data set noisier.
- Filter devices detected at only one sensor. As we are interested in movements of crowds devices that do not move or that have only been detected once and never seen again bring no information about the behavior of crowds, they just create noise in the data.

Table 2 Data set characteristics

	Total number of detections	**Number of Sensors**	**Days of Interest**
Arnhem	2472380	5	1
Assen	11860349	27	3

To test these filters we used two different data sets, one from the city of Arnhem that we produced using our own sensors and one from the city of Assen that was provided for us.

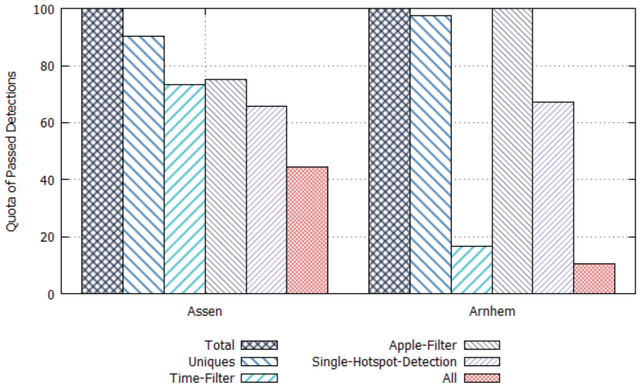

Fig. 5 Centralized Filter Results

In Table *2* we can see of the characteristics of the two data sets, we can observe how the two traces are very different in the number of detections and the number of hotspots that gathered these detections, the interest data also spans over different time frames. In Arnhem's case we were interested in the Living Statues Festival and for the case of Assen the interest was a TT Festival, these 2 festivals each attracted about a hundred thousand visitors to these 2 very similar cities.

By using the filters onto these two distinct data sets we obtained the results in Fig. *5*. We can observe how the initial data set has been reduced and how applying all four filters reduces the data set even more. The Arnhem data set has been reduced to 10% of its original size and the Assen one to 44% of the original size. The large difference between the two is given by the number of data points outside the interest time, in the Assen data set there were a few extra hours of detection data while Arnhem data set had a few days of data.

V. Future Work

In showing how filters affect the size of the data set and the effectiveness they show we managed to prove the need for these filters in applications of these type and others. Other filters still need to be researched that could reduce the data set even further.

After filters data aggregation techniques could further reduce the data set by merging several detections or movements into one. There is little research in how this data should be aggregated and what the limitations are as well as how effective such an aggregation would be in reducing the final data set.

VI. Conclusions

In this paper we discussed systems that use Wi-Fi packet detections to track crowds of people inside a city area. We showed through examples of other similar projects why we expect the number of detections in the resulting data sets would continue to grow and reach sizes that make processing difficult and could increase the execution time to levels where the result would be obtained too late.

For these systems we proposed and explained several filtering methods and showed the effect they have on the data and the reduction they manage to achieve. Three filters work at sensor level while the others require knowledge of the entire data set or of the results of multiple sensors. We showed what effects these filters have on real life data obtained from several different sources and how effective they are in minimizing the data set. The filters we proposed are mostly targeted at detections obtained from Wi-Fi packet sensors, but some of them can be generalized to detection registering method.

We managed to show the need for such filters in both making future similar projects a more acceptable challenge as well as opening the way into research of more possible filters.

Acknowledgment

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/187/1.5/S/155536, and national project MobiWay, Project PN-II-PT-PCCA-2013-4-0321.

We would like to thank Roel Schiphorst from BlueMark Innovations for providing us with the Aseen data set and to Spyros Voulgaris and Claudio Martella for their support and suggestions during this project.

REFERENCES

[1] A. Schmidt, "Low-cost Crowd Counting in Public Spaces".

[2] B. Bram, A. Barzan, P. Quax and W. Lamotte, "WiFiPi: Involuntary tracking of visitors at mass events," in *World of Wireless, Mobile and Multimedia Networks* , 2013.

[3] K. Eftychia, R. Šilerytė, M. Lam, K. Zhou, M. v. d. Ham, E. Verbree and S. v. d. Spek, "Passive WiFi Monitoring of the Rhythm of the Campus," in *AGILE*, Lisbon, 2015.

[4] Ruiz-Ruiz, A. J., H. Blunck, T. S. Prentow, A. Stisen and M. B. Kjaergaard, "Analysis methods for extracting knowledge from large-scale WiFi monitoring to inform building facility planning," in *Pervasive Computing and Communications* , 2014.

[5] I. Sibren, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky and W. Willinger, "Human mobility modeling at metropolitan scales," in *international conference on Mobile systems, applications, and services*, 2012.

[6] Chan, A. B., Z.-S. J. Liang and N. Vasconcelos, ". "Privacy preserving crowd monitoring: Counting people without people models or tracking."," in *Computer Vision and Pattern Recognition*, 2008.

[7] R. D., D. S., F. C. and Sridharan, "Crowd counting using group tracking and local features," in *Advanced Video and Signal Based Surveillance* , 2010.

[8] Aggarwal, J. K. and Q. Cai, "Human motion analysis: A review," *Computer vision and image understanding* , vol. 73, no. 3, pp. 428-440, 1999.

[9] Z. Beibei, D. N. Monekosso, P. Remagnino, S. A. Velastin and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications,* vol. 19, no. 5-6, pp. 345-357, 2008.

[10] C. Chuan-Yu, W.-H. Tung and J.-S. Wang, "A crowd-filter for detection of abandoned objects in crowded area," in *International Conference on Sensing Technology*, 2008.

[11] Andrade, E. L., S. Blunsden and R. B. Fisher, ""Modelling crowd scenes for event detection."," in *International Conference Pattern Recognition*, 2006.

[12] Loscos, Celine, D. Marchal and A. Meyer, "Intuitive crowd behavior in dense urban environments using local laws," in *Theory and Practice of Computer Graphics*, 2003.

[13] B. A., N. S., C. S. and D. Manocha, "Densesense: Interactive crowd simulation using density-dependent filters," in *Symposium on Computer Animation*, 2014.

[14] K. Huber, M. Schreckenberg and T. Meyer-König, "Models for crowd movement and egress simulation," *Traffic and Granular Flow,* pp. 357-372, 2005.

[15] E. Frans, A. Keränen, J. Karvo and J. Ott, "Working day movement model," in *ACM SIGMOBILE workshop on Mobility models*, 2008.

[16] K. Dmytro, C. Boldrini, M. Conti and A. Passarella, "Human mobility models for opportunistic networks," *Communications Magazine,* vol. 49, no. 12, pp. 157-165, 2011.

[17] S. P., S. A., G. R. and L. S, "Tracking Human Mobility using WiFi signals.," in *arXiv preprint arXiv:1505.06311.*, 2015.

[18] W. Yan, J. Liu, Y. Chen, M. Gruteser, J. Yang and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *international conference on Mobile computing and networking*, 2014.

[19] S. Depatla, A. Muralidharan and Y. Mostofi, "Occupancy Estimation Using Only WiFi Power Measurements," *JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,* vol. 33, no. 7, p. 1381, 2015.

[20] X. Wei, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu and Z. Jiang, "Electronic frog eye: Counting crowd using wifi," in *INFOCOM*, 2014.

[21] K. Yungeun, H. Shin and H. Cha, "Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem," in *PerCom*, 2012 .

[22] R. Valentin and M. K. Marina, "Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting," in *International Conference on Indoor Positioning and Indoor Navigation* , 2013.

[23] H. F., Z. Y., Z. Z., W. M., F. Y. and Z. Guo, "WaP: Indoor localization and tracking using WiFi-Assisted Particle filter," in *Local Computer Networks* , 2014.

[24] W. Yan, J. Yang, H. Liu, Y. Chen, M. Gruteser and R. P. Martin, "Measuring human queues using WiFi signals," in *international conference on Mobile computing & networking*, 2013.

[25] O. Chris, J. Jiang and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *ACM SIGMOD international conference on Management of data*, 2003.

[26] M. A. B. M. and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *10th ACM conference on embedded network sensor systems*, 2012.

[27] Apple , "Apple - Privacy - Privacy Built In," Apple , 17 07 2015. [Online]. Available: http://www.apple.com/privacy/privacy-built-in/. [Accessed 17 07 2015].