Discovering Hotspots: A Placement Strategy for Wi-Fi based Trajectory Monitoring within Buildings

Lorenz Schauer
Mobile and Distributed Systems Group
Ludwig-Maximilians Universität
Munich, Germany
Email: lorenz.schauer@ifi.lmu.de

***Abstract*—In the last decade, Wi-Fi based trajectory monitoring has gathered high interest in the scientific and commercial world, due to the increased usage of Wi-Fi capable mobile devices. A lot of work can be found where monitor nodes are placed in an area of interest capturing Wi-Fi signals from passing phones. However, the deployment of such nodes is often inefficient, expressed by a low ratio between monitored trajectories and the amount of installed nodes. Hence, finding an optimal setting of node positions is an essential and challenging task. In this paper, a systematic solution for this variant of the NP-hard art gallery problem is investigated. The idea is to set monitor nodes only on places (hotspots), where most of the human paths can be tracked. For the discovery of such hotspots, three novel approaches are presented working on simulated user traces based on an extended pathway mobility model, and a given plant layout. The results of each approach are evaluated in terms of quality for Wi-Fi based trajectory monitoring using different parameters and settings. The evaluation indicates that the proposed methods show different potentials and limitations. Overall, they return a reliable setting of hotspots compared to a completely random selection of an equal amount of node positions and, thus, they serve as a systematic and sophisticated placement strategy.**

Keywords*—Hotspot Discovery; Node Placement; Art gallery; Mobility Model; Trajectory Clustering

I. Introduction

The immense diffusion of modern mobile devices with integrated sensors and several communication interfaces have led to an increased usage of Wi-Fi as de facto wireless communication standard. In many situations and locations of daily life, e.g. at work, at home, in shopping malls or in other public places, wireless networks are available, offering Internet access and local services to people. In order to use these services in an ubiquitous way, Wi-Fi enabled mobile devices periodically scan their vicinity for known or free networks and try to connect to them automatically. Such 802.11 active scans leak unencrypted information to the surroundings, e.g the device-specific MAC-address, and can easily be captured by any Wi-Fi card set into monitor mode. A lot of work in literature can be found where an infrastructure of several Wi-Fi monitor nodes has been deployed into an area of interest to gather crowd data [4], social relationships [3], or estimate pedestrian flows [16], and track human trajectories [15].

However, the deployment strategy of such nodes is never further explained. E.g., the authors in [4] express that 15 monitor nodes "were placed at strategic locations", but it is not discussed if the same or more mobile devices could have been tracked with a smaller amount of nodes placed at other locations. In other words, the ratio between the amount of tracked devices and the number of used monitor nodes is not investigated in terms of efficiency, and a random selection of node positions does not return the best solution. Adequate setup strategies and approaches for finding optimal positions within a given scenario are still missing. Due to the fact, that each additional monitor node leads to higher installation and maintenance cost and causes more overhead in communication and data analysis, such approaches would help to save money by reducing the amount of required nodes without decreasing the amount of trackable devices.

In this paper, we focus on this optimization problem for a Wi-Fi based trajectory monitoring infrastructure within large public buildings. Such an infrastructure captures Wi-Fi signals from passing phones at various places in order to reproduce the original trajectories, represented as a series of coordinates, taken by the visitors of the building. For this purpose, we simulate various visitors moving through a bitmap representation of a floor plan on an underlying extended pathway mobility model. Based on these simulations, we try to find the best locations for the placement of monitor nodes in a given building. In order to reach this goal, we present and implement three approaches returning potential node positions: A grid-based method, a density clustering method using DBSCAN [8], and a trajectory clustering method using the TRACLUS [14] algorithm. The positions returned by these methods are evaluated in terms of quality for Wi-Fi based trajectory monitoring and are compared against a completely random selection of the same amount of nodes.

The goal is, that a maximum amount of human trajectories can be tracked by a minimum amount of monitor nodes. Intuitively, the best location for a monitor node is found when it's range covers those areas which are frequented by a maximum amount of people. In a heatmap, these areas would be represented as "very hot" and, hence, the best and optimal positions for Wi-Fi monitors are called "hotspots" in this context. The contributions of this paper can be summarized as follows:

- A variant of the well-known art gallery problem is presented
- An extension of the pathway mobility model [2] is introduced in order to simulate visitors inside buildings
- Three novel approaches for finding the most fre-

DOI: 10.1109/IntelliSys.2015.7361169 $33.00 ©2015 IEEE

quented areas are presented and evaluated on different environments and for various input parameters

- An alternative method for the partitioning phase of the TRACLUS algorithm is presented, resulting in a more precise segmentation phase for human trajectories within buildings and reducing the computational effort for clustering, due to less segmentation points.

The paper is organized as follows: Section II gives a brief overview of related work. Some preliminaries and the problem statement are presented in Section III. Based on these definitions, the methodology is described in Section IV introducing the used models and three hotspot discovery approaches. The evaluation of these approaches is presented in Section V and, finally, Section VI concludes the paper and gives hints on future work.

II. Related Work

This paper is related to current research topics, such as hotspot discovery, Wi-Fi based trajectory estimation, and strategies for sensor placement. Thiagarajan et al. [17] analyze collected data from Wi-Fi captures and estimate both human trajectories and travel times in street networks. Road segments with lots of traffic are named as hotspots which are discovered when a remarkable high travel time is observed. Hoteit et al. [11] also use the term of hotspots for the most crowded regions in cellular networks. Based on data from mobile phone activities, the authors estimate human trajectories using different interpolation methods and discover hotspots with a median error lower than 7%. Ahmed et al. [1] define hotspots as a location with a user density higher than a predefined threshold. Based on this definition, an approach is presented returning all of the existing hotspots based on indoor tracking data. In comparison to these works, we define hotspots as the center of regions which are passed by a maximum amount of people. This is similar to the hot route discovery problem, trying to find the most frequently traveled routes [6], [19].

Several real word experiments can be found deploying a certain amount of Wi-Fi monitor nodes in an area of interest in order to estimate human trajectories [9], [15]. However, adequate placement strategies are not discussed further and approaches for finding the best node locations are still missing. In general, finding optimal locations for sensor nodes is a big challenge and a lot of research is done in this field. Different strategies and techniques for node placement in wireless sensor networks are presented in literature [18]. An optimization is introduced by Krause et al. [12], [13], trying to find the k best locations for sensor nodes in an area of interest where a finite set of possible node locations exists. An optimized solution for visual sensor placement is presented by Gonzalez [10], or Bottino and Laurentini [5]. Both works present an algorithm in order to solve a variation of the art gallery problem which is similar to the problem statement of this paper. However, our approach focuses on discovering hotspots for efficient Wi-Fi based trajectory monitoring within buildings, and to the best of our knowledge, this has not been investigated so far.

III. Definitions and Problem Statement

The overall goal is to monitor a maximum amount of human trajectories with a minimum amount of deployed nodes. Obviously, this goal has two opposed properties: The minimization of deployed nodes and the maximization of trackable human paths. In order to reach this goal, the following definitions are introduced before presenting the problem statement:

A trajectory t is a time-series of location records representing a human path P_i. In our context, P_i is represented as a sequence of two-dimensional points p, denoted as $P_i = \{p_1, p_2, ..., p_n\}$. Inside a building, a person moves within walkable regions along a spatial network of floors, rooms, halls, etc.

According to [14], each trajectory t can be partitioned at characteristic points, where the behavior of t changes significantly. We only consider the direction of a person's movement for choosing characteristic points, due to the fact that a person usually walks straight along shortest paths and suddenly changes the walk direction on certain locations, e.g. corners, doors, entries, etc, rather than walking completely random. Hence, a characteristic point is defined as follows in this context:

Definition 1 (Characteristic Point): A point $p \in P_i$ is marked as characteristic point $cp \in P_i$ when the direction vector ν_t changes at p with an angle $\alpha \geq \triangle$.

On the basis of such characteristic points, we define sub-trajectories:

Definition 2 (Sub-trajectory): A sub-trajectory $\tau \subseteq t$ represents a partition of a trajectory t and is denoted as a line segment between two successive characteristic points $cp_n cp_m$ with $n < m$.

In order to monitor a complete trajectory as accurate as possible, it becomes necessary to be able to track a maximum amount of its sub-trajectories:

Definition 3 (Trackable Sub-trajectory): A sub-trajectory τ is trackable, if there is at least one point $p \in \tau$ located within the coverage range r_m of at least one monitor node m, formally desired as:

$$\exists p \in \tau . \exists m \in M : dist(p, m) \leq r_m \tag{1}$$

where $\mathrm{dist}(p, m)$ is the Euclidean distance between the monitor node m and one point p of the sub-trajectory τ.

In other words, a sub-trajectory can be detected by Wi-Fi captures if it passes at least one monitor node's coverage range. As an example, Figure 1 shows trackable and non-trackable sub-trajectories of a sample path from start S to destination D. The last sub-trajectory before reaching D is not covered by any Wi-Fi monitor's range and, thus, it is not trackable.

Formally, the given optimization problem is closely related to the well-known and NP-hard art gallery problem where a minimum set of guards inside a polygonal art gallery has to be found in order to observe the whole area. Some modifications have to be made in order to respect the properties of Wi-Fi monitors which can be seen as guards: Instead of observing all line-of-sight points inside the polygonal area, a Wi-Fi monitor's range is limited, due to physical restrictions. Furthermore, Wi-Fi monitor nodes listen to their surroundings, rather than watching it and, thus, they can also track mobile devices behind walls and other obstacles within their coverage

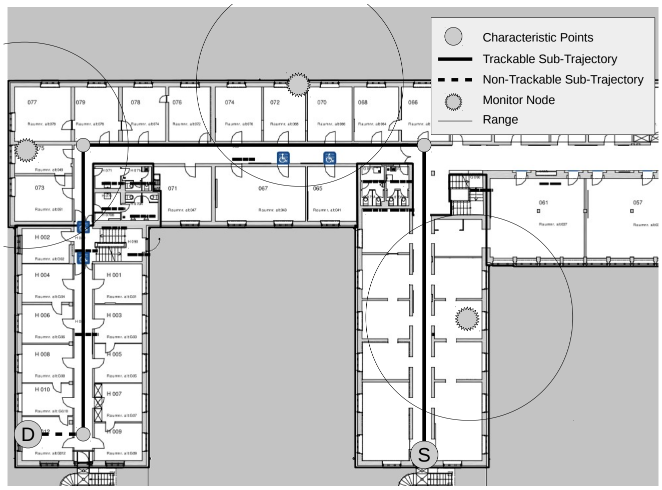


Fig. 1. Example of trackable and non-trackable sub-trajectories

range. According to [10], and with respect to the proposed modifications, the problem statement is defined as follows:

Definition 4 (Problem Statement): Given a polygonal layout $\Gamma \subset \mathbb{R}^2$ and a set of human trajectories T. Find a minimum set of monitor node (guard) locations $G = \{g_{m_1}, g_{m_2}, ..., g_{m_n}\}$ with coverage range $r_{m_1}, r_{m_2}, ..., r_{m_n}$ inside of Γ, such that a maximum set of sub-trajectories τ is trackable from at least one point in G according to Definition 3.

In order to solve this problem, monitor nodes should be deployed at locations where most people pass by, and thus, where the most sub-trajectories are trackable. As already mentioned, these locations are called "hotspots". The following section introduces a methodology with three different approaches for an automatic detection of such hotspots.

IV. Methodology

The algorithms presented in this section are based on human trajectories created by simulated persons moving inside buildings. For the building's representation, a simple, and metrical environment model is used.

A. Environment Model

The basis of our environment model is an image of a typical plant layout, as shown in Figure 2(a). The image is preprocessed and divided into walkable and non-walkable regions. All other information shown in the original image, such as doors, room numbers, signs, etc., are removed in this step. Furthermore, the outdoor parts of the given plant layout have to be marked as walkable or non-walkable, depending on the requirements. The original image is then converted into a binary image, and stored as a bitmap representation. This leads to a very small image size and makes walkable and non-walkable regions easily distinguishable, as shown in Figure 2(b).

White regions encoded with 1 are walkable, and black regions encoded with 0 are non-walkable areas. Thus, people are able to move in any direction and through any connected areas of white pixels with respect to the coordinate system of the given plant layout. As a last step, rooms, floors, and entrances are labeled which is important for the mobility model introduced in the sequel.

(a) Original image (b) Bitmap of preprocessed image

Fig. 2. A bitmap of a typical plant layout used as environment model

B. Mobility Model

For an adequate simulation of persons moving within the proposed environment model, we present a modification of the pathway mobility model taking the probability for choosing a certain destination into account.

In few words, the pathway mobility model is based on the frequently used random waypoint model, but it considers geographic constraints of the simulation field which usually exist in real life environments, e.g. pedestrians cannot walk through walls and are bounded to move along floors. In the pathway mobility model, nodes are moving in a pseudo-random fashion on predefined pathways. Every node is randomly placed at the start of the simulation and chooses a destination in the simulation field. It moves towards the desired destination on the shortest path along the walkable space. After reaching the destination, a node stays at this position for a certain period of time and then again, a new destination is chosen randomly. This is repeated until the simulation ends [2].

For our purpose, the pathway mobility model fits in the sense, that visitors of a building usually chose a certain destination $d \in D$ and try to reach it on the shortest path, while their movement is restricted to the walkable regions of the building. When a person arrives at d_i, the next destination d_{i+1} is chosen which may be another room or, finally, the way out. However, for a more realistic simulation of visitors in public buildings, we present the following modifications and extensions to the pathway mobility model:

- Each visitor $v \in V$ of a certain building enters and exits the building through an entrance door $e \in E$ with the probability $P_v(e)$. This allows for a differentiation of various entrance doors, e.g. main entrance and back doors. Note, that a person does not have to enter and exit the building through the same door.
- Instead of placing a person randomly into the building, each visitor is located at an entrance door at the beginning, expressed by $start = e \in E$.
- In realistic scenarios, a person doesn't choose a destination randomly. A destination is always connected with an objective which is usually achieved in a certain room $r \in R$, e.g. go to the lecture room in order to participate at a lecture, or go to the cafeteria in order to have lunch. Therefore, a destination within

Require: $environment\_model$
V:= number of Visitors
for $(v = 1 : V)$ **do**
$D := number\_of\_Destinations$
/*choose entrance e with probability $P_v(e)$*/
$start = getEntranceDoor()$
for $(d = 1 : D)$ **do**
if $(d == D)$ **then**
/*choose exit e with probability $P_v(e)$*/
$d = getEntranceDoor();$
else
/*choose destination d with probability $P_v(d)$*/
$d = get\_next\_Destination();$
end if
/*compute shortest path between start and d*/
$path = compute\_dijkstra(start, d);$
/*Store $path$ for visitor v*/
$savePathPerVisitor(path, v);$
$start = d;$
end for
end for

Fig. 3. Simulation algorithm of the proposed mobility model

a building is usually a room, rather than a floor or staircase. Thus, in our mobility model a destination is defined as $d \rightarrow room_x$.

- Only the last destination d_{last} of a visitor must be an exit door in order to leave the building: $d_{last} \rightarrow e$, and e is chosen with the probability $P_v(e)$.
- Any destination is chosen by a visitor at a certain time t with the probability $P_{v,t}(d)$. This modulation allows for a more realistic simulation of choosing destinations with respect to the time, a decision is made. E.g., a lecture room is visited more before lessons, and the cafeteria is visited more during lunch time. Some rooms are chosen with a higher probability than others at the same time, e.g, the main lecture room compared to smaller ones.

The simulation algorithm for the proposed mobility model is depicted in Figure 3. In summary, we create a certain amount of visitors V and let them move on shortest paths from start to the particular destination through a given building. Each visitor chooses a number of destinations w.r.t the given probabilities and leaves the building through an entrance door as last destination.

C. Hotspot Discovery Approaches

As already mentioned, hotspots in this context represent a set of locations inside a given building where Wi-Fi monitor nodes should be placed in order to track human trajectories in an efficient way. On the basis of the proposed simulations, three approaches are introduced in the sequel, allowing for a systematic discovery of such hotspots.

1) Grid-Based: As a first step, the grid-based approach creates a heatmap representation of the simulated trajectories. Such a heatmap illustrates how often a certain location in the building has been visited. Every time a person passes a pixel coordinate of the bitmap, the corresponding heat value is increased by one. For a better illustration, Figure 4 represents such a heatmap of 1000 simulated persons each choosing five destinations in a main building of a big university.

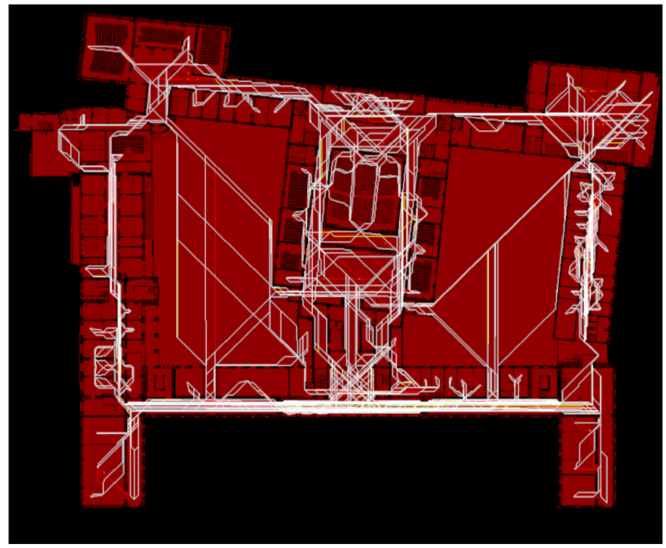


Fig. 4. Heatmap of 1000x5 simulated paths in a university building.

In the second step, a square grid with constant square width w_s is placed over the heatmap, and the pixel sum of each square s_Σ is stored together with the pixel coordinates of the corresponding square center c_s. This results in a list $L \rightarrow < c_s, s_\Sigma >$ indicating the frequency of how often a square has been visited by a person. The list entries c_s and s_Σ depend on w_s. In our case, this parameter is set as $w_s = m_r \cdot \sqrt{2}$ and, hence, a monitor node's coverage radius m_r serves as the minimum bounding circle of a square. Thus, every pixel coordinate inside the building can be observed, when monitor nodes are only allowed to be placed at the square centers of the grid.

As a last step, hotspots are determined out of all square centers $c_s \in L$. Formally, a square center c_s is marked as hotspot if $s_\Sigma \geq \varepsilon \cdot \arg\max_{s_\Sigma \in L}\{s_\Sigma\}$, with $0 \leq \varepsilon \leq 1$. In other words, if the pixel sum of a square is greater than a certain percentage ε of the most frequented square, the location of its center is marked as hotspot. Obviously, the result depends directly on ε which has to be chosen individually, according to the given requirements.

Note, this simple approach only considers how often a square is frequented by visitors, rather than respecting the course of trajectories. Thus, essential points of human paths, e.g. where significant direction changes occur, may not be covered by any node if the pixel sum of the particular square is low and, hence, important information for tracking human movements is getting lost. In order to solve this problem, we present the following approach focusing on direction changes of human trajectories.

2) Density-Based: This approach is called density-based, because it uses the DBSCAN algorithm [8] in order to perform a density based clustering of characteristic points. Centers of discovered clusters are then marked as hotspots. The algorithm of this approach is shown in Figure 5.

Require: A set of trajectories $T = \{t_1, t_2, ..., t_i\}$
Parameters: ε, *minPts*, and ϵ

/*Step 1: Find characteristic points/*
for all $(t_i \in T)$ **do**
$index = 1$;
Add $p_1 \in t_i$ to the set CP_i of characteristic points;
$len = 1$;
$v_1 = createVector(p_{index}, p_{index+len})$;
while $((index + 2 \cdot len) \leq length(t_i))$ **do**
$v_2 = createVector(p_{index+len}, p_{index+2 \cdot len})$;
if $(vectorAngle(v_1, v_2) \geq \frac{1}{3}\pi)$ **then**
Add $p_{index+len}$ to CP_i;
$v_1 = v_2$;
end if
$index$++;
end while
Add last point of t_i to CP_i;
/*Use Douglas-Peucker to smooth CP_i*/
$SP_i = performDouglasPeucker(CP_i, \varepsilon)$;
Add SP_i to result set R;
end for

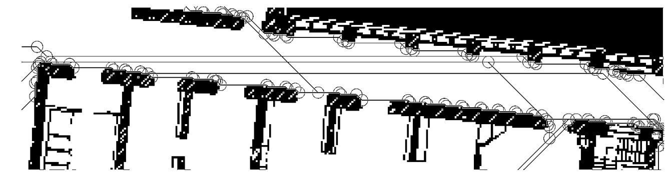
/*Step 2: Get clusters of characteristic points*/
$Cl = DBSCAN(R, minPts, \epsilon)$;
return $mean(cl_i \in Cl)$; /*Cluster centers*/

Fig. 5. Algorithm of density-based approach for hotspot discovery

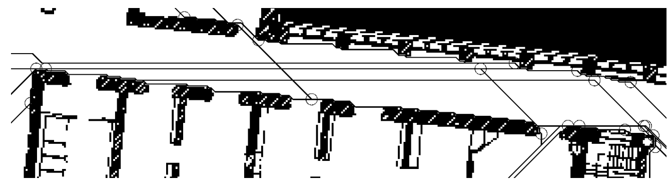
As a first step, the characteristic points are determined and extracted from the simulated trajectories. According to Definition 1, a characteristic point is found when the direction vector changes with $\alpha \geq \triangle$. As shown in Figure 5, we set $\triangle = \frac{1}{3}\pi$. Thus, points are marked as characteristic, if the walk direction changes with at least $60°$. Intuitively, this is a reasonable value for indoor scenarios where persons usually change their walk direction rapidly with $\alpha \approx 90°$, e.g. when turning around a corner, or entering a room.

In our simulations, however, we also observe particular direction changes of the trajectory with $\alpha \approx 90°$ when a person walks next to an obstacle or a wall. This is caused by computing shortest paths on bitmaps where each white pixel represents a possible location, and a simulated path can run along a wall pixel per pixel, as it is shown in Figure 6. Due to this fact, many characteristic points are extracted by our algorithm which do not represent a real direction change of a person, depicted as little circles in Figure 6(a). Hence, these points should not be marked as characteristic and have to be removed from the set of CP_i. For this purpose, we use the line simplification algorithm of Douglas-Peucker [7], and smooth each trajectory of characteristic points CP_i with a small ε value. As an example, the results of this step are shown in Figure 6(b) using $\varepsilon = 7$. The remaining characteristic points SP_i of each trajectory are added to the result set R.

In the second step, R is given as input parameter to the DBSCAN algorithm in order find density based clusters of characteristic points. Note that results of DBSCAN depend significantly on the required input parameters *minPts* and ϵ determining the minimal number of points required to form a dense region within an ϵ neighborhood radius. In our case,



(a) Before smoothing



(b) After using Douglas-Peucker

Fig. 6. Extracted characteristic points on simulated paths

this radius is set w.r.t. a monitor node's coverage range as upper bound: $\epsilon \leq r_m$. Furthermore, we restrict *minPts* with $1 \leq$ *minPts* $\leq |T|$, so a cluster has to contain less or equal points than the amount of given trajectories. A higher value for *minPts* leads to very big clusters which are not practicable for our purpose. The parameter *minPts* allows to react on distinct situations or individual requirements for node placement: if $minPts = 1$, the most clusters are detected, and lead to a maximum amount of hotspots, and also to a higher probability for tracking all visitors. For *minPts* $= |T|$ less clusters (down to one single cluster) are found leading to a minimum amount of discovered hotspots, but also to a minimum number of trackable trajectories.

Finally, the center of each cluster is computed as the mean of all cluster points. The set of all center points $C = \{C_{cl_1}, C_{cl_2}, ..., C_{cl_m}\}$ is returned and represents the set of discovered hotspots.

Considering the time complexity, the proposed algorithm shows a linear complexity $O(n)$ for the discovery of characteristic points, where n is the number of points on a trajectory t_i. The complexity of Douglas-Peucker is $O(n^2)$ in worst case, with $n = |CP_i|$ denoting the number of discovered characteristic points of a trajectory t_i. The complexity of DBSCAN is also $O(n)$, with $n = |R|$ in this case.

In comparison to the grid-based method, this approach considers the course of trajectories focusing on rapid direction changes. In the following section, an extension to this method is presented, considering more characteristics of sub-trajectories for a reliable tracking of human paths.

3) Trajectory-Based: The trajectory-based approach uses the TRACLUS algorithm [14] in order to find representative trajectories of clusters. Such clusters are formed through a density-connected set of similar sub-trajectories. Due to the usage of Dijkstra for shortest path routing, we only consider characteristic points according to Definition 1 for the partitioning of trajectories. Hence, we use the presented algorithm for finding characteristic points from the previous section, rather than computing the MDL cost for each point of a line segment, as described in [14]. The proposed modification is suitable for our purpose resulting in a more precise and less complex partitioning phase for human trajectories within buildings.

Furthermore, less segmentation points are returned reducing the computational effort of TRACLUS' grouping phase.

This phase performs a density-based clustering of line segments using principles of DBSCAN. Again, two parameters *minLns* and ϵ are required in order to build clusters of line segments. A cluster is then formed by a density connected set around core line segments which have a minimum amount of lines *minLns* within their ϵ-neighborhood. Like before, we define $\epsilon \leq r_m$ and $1 \leq \mathit{minLns} \leq |T|$, so we focus on clusters containing a minimum amount of *minLns* sub-trajectories within a monitor's coverage range. This is very useful for our purpose, where monitor nodes should only be placed in areas where many people pass by.

As a last step, the TRACLUS algorithm introduces a method for constructing representative trajectories RT. For the discovery of hotspots, we highly benefit from this step, because each representative trajectory $rt_i \in RT$ represents the characteristic movement of all sub-trajectories of the corresponding cluster cl_i. Important parts of human paths are described by the representative trajectories and, hence, they are required to be trackable according to Definition 3. A basic solution would be to install one monitor node m for each $rt_i \in RT$ such that Equation 1 is satisfied. However, this solution would require too many nodes, because some representative trajectories might be close together and could be covered by one single node. Hence, we use DBSCAN with $\epsilon = r_m$ and $\mathit{minPts} = 1$ in order to group start and endpoints of representative trajectories which are located in a range of one monitor node. The centers of returned clusters are marked as hotspots. Note that noisy points of this step are not removed. They are rather considered as hotspots, because these points represent particular start or endpoints which are not in the vicinity to others and cannot be clustered by DBSCAN.

V. Evaluation

In this section, the presented approaches are evaluated in different environments and for a various amount of simulated paths. First of all, we investigate the effect of required input parameters, e.g. ε, *minPts*, or *minLns* for each approach. Afterwards, the returned hotspots are evaluated against a random selection of monitor node positions with respect to the amount of trackable sub-trajectories.

A. Settings

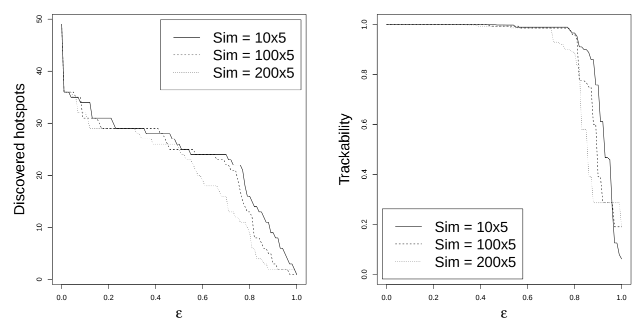
Four plant layouts are used for evaluation, representing different characteristics and probabilities for choosing a certain destination:

1) The principle building of a university, where lecture halls are selected with a higher probability according to their capacity, and entrances are used on the basis of their importance, e.g. the main entrance is used with a higher probability than side entrances. The corresponding plant layout is depicted in Figure 7(a) representing the biggest environment of our evaluation.
2) An office environment, where offices are visited with the same probability and special rooms, e.g. conference room, cafeteria, toilette, etc., can be visited more often depending on the scenario. The environment is illustrated in Figure 7(b).
3) A town hall, where a lot of people are attended each day. Thus, the probability of choosing the waiting and public office is much higher than going to the mayor's office. Figure 7(c) depicts the town hall.
4) A hospital floor, with one operating room and several dorms. Again, the main entrance is chosen with a higher probability than both side entrances and the dorms are selected according to the number of beds. The hospital is illustrated in Figure 7(d).

On each of these environments, we perform 20 simulations creating a small amount of 10 up to a higher amount of 200 persons choosing 5 destinations inside the corresponding building. As a simplification, and, due to the fact that we have no adequate Wi-Fi propagation model for each of the environments, the coverage range of a monitor node is set to $r_m = 25$ meter which is a realistic value for indoor scenarios.

B. Grid-Based Approach

As described in Section IV-C1, a square center is marked as hotspot, if the pixel sum of the square is greater or equal than a certain factor ε of the most frequented square. For evaluation, we vary ε from 0.00 to 1.00 for each environment. Furthermore, we investigate the influence of the simulations and create 10, 100, and 200 persons selecting 5 destinations within each building. The results for the university building as the largest environment are depicted in Figure 8(a). Obviously,

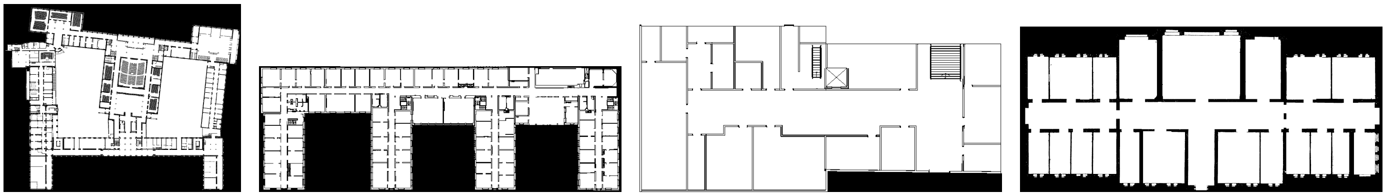


(a) Hotspot discovery (b) Trackable trajectories

Fig. 8. Effect of ε varying the amount of simulated persons at the university environment

the amount of square centers which are marked as hotspots increases with decreasing ε. For $\varepsilon = 1.0$, one hotspot is returned denoting the most frequented square. On the opposite, for $\varepsilon = 0.0$ every square center is marked as hotspot. It is shown, that in case of less simulations, more squares are marked as hotspots for the same ε value, particularly for $\varepsilon > 0.5$. This is evident, because the pixel sums of squares are not as different as in case of more simulations and, hence, more square centers are marked as hotspots when decreasing ε.

As next step, we evaluate how many sub-trajectories are trackable according to Definition 3 when placing monitor nodes at the discovered hotspots. More precisely, we consider the length of these trackable sub-trajectories, rather than just their amount. The length can be seen as a weight for each trackable sub-trajectory, leading to the fact, that it is more



(a) University (b) Office (c) Town hall (d) Hospital

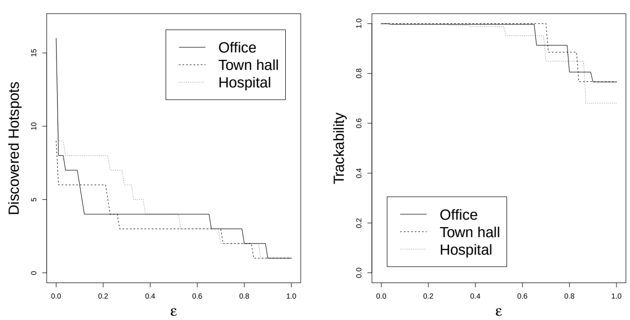
Fig. 7. Bitmap representations of plant layouts used as test environments for evaluation

important to track longer sub-trajectories. This is comprehensible for our purpose. In order to evaluate the length of trackable sub-trajectories, the term of trackability is introduced as follows:

Definition 5 (Trackability): The trackability is defined as the ratio between the length of trackable sub-trajectories and the total length of all trajectories in the used environment.

According to this definition, Figure 8(b) depicts the results for the university environment w.r.t ε. It can be observed, that the trackability is almost irrespective of the amount of simulations for $\varepsilon \leq 0.7$, where nearly the complete length of sub-trajectories is trackable. For $\varepsilon > 0.7$ the trackability is drastically decreasing in case of all simulations. Taking the amount of hotspots into account, an acceptable trade-off between a minimum amount of required nodes and a maximum trackability is observed for $\varepsilon \approx 0.75$ within these settings.

Similar results are obtained for the other environments, depicted in Figure 9 when using 200 simulated persons. Due



(a) Hotspot discovery (b) Trackable trajectories

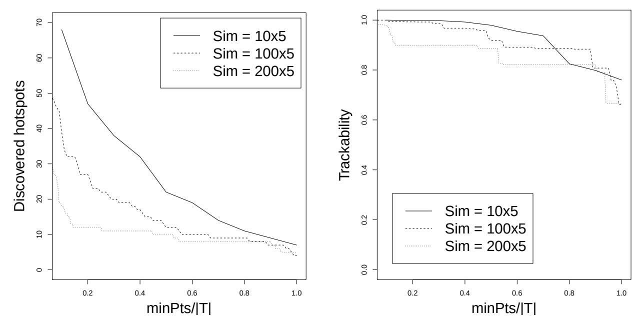
Fig. 9. Effect of ε using 200 persons selecting 5 destinations at the other environments

to the fact, that these environments are much smaller, a lower amount of nodes suffices to cover all sub-trajectories. While the university environment requires about 18 nodes for the high amount of simulations, only 3 to 4 nodes are needed in the other buildings for a complete coverage of trajectories setting $\varepsilon \leq 0.6$. On the other hand, one single node placed at one discovered hotspot already tracks more than 70% in these buildings and, thus, additional discovered hotspots do not increase the amount of trackable sub-trajectories as drastically as in case of the university environment. However, for $\varepsilon > 0.6$ we also observe that the trackability decreases significantly.

C. Density-Based Approach

Due to the usage of DBSCAN for clustering characteristic points, the results of this approach depend on the used input parameters, such as ϵ and *minPts*. Previous tests for the used buildings show that in case of $\epsilon = \frac{1}{4} r_m$ the found clusters indicate an adequate size compared to a monitor node's coverage range. With $\epsilon = r_m$ the clusters are too large, and for smaller ϵ many clusters of a very small size with lots of noise have been found.

Based on these findings, we set $\epsilon = \frac{1}{4} r_m$ and investigate the effect of *minPts* on both the amount of discovered hotspots and the corresponding trackability. According to the restrictions made in Section IV-C2, we vary *minPts* from 1 to $|T|$. Like for the previous approach, this is done for a low, middle, and a higher amount of performed simulations. Figure 10 depicts the results for the university environment. For a better comparability and illustration, *minPts* is normalized by the maximum value $|T|$.



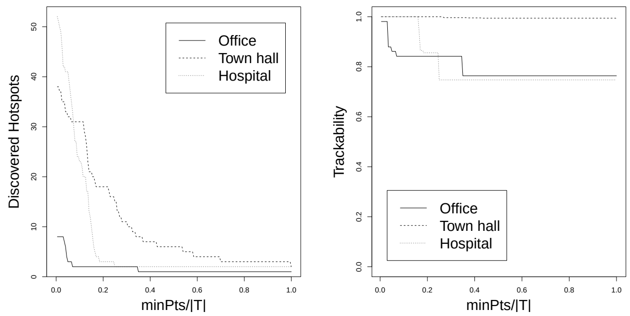
(a) Hotspot discovery (b) Trackable trajectories

Fig. 10. Effect of *minPts* varying the amount of simulated persons at the university environment

As before, Figure 11(a) shows that a higher amount of simulations leads to a lower number of discovered hotspots for the same value of *minPts*. This is due to the nature of DBSCAN. More paths lead to more characteristic points which lead to a higher density and, thus, to a higher probability for finding less but larger density connected sets. Therefore, we receive less clusters and, thus, less hotspots. Another observation is, the higher the value of *minPts*, the lower the amount of found clusters and of discovered hotspots, respectively.

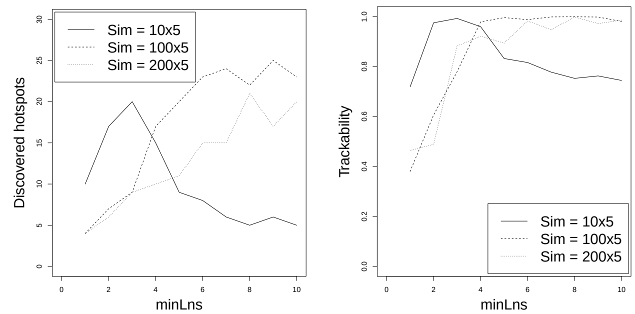
As depicted in Figure 11(b), more than 65% of the total trajectory length is trackable using more than 7 nodes with *minPts* $= |T|$. This indicates an improvement to the previous approach. However, in order to track the complete length of sub-trajectories, more than 28 nodes are required, indicating a worsening to the grid-based method.

Figure 11 shows the results which are obtained for the remaining environments using 200 simulated persons. It can be observed, that the amount of hotspots remains almost stable



(a) Hotspot discovery (b) Trackable trajectories

Fig. 11. Effect of *minPts* using 200 persons selecting 5 destinations at the other environments



(a) Hotspot discovery (b) Trackable trajectories

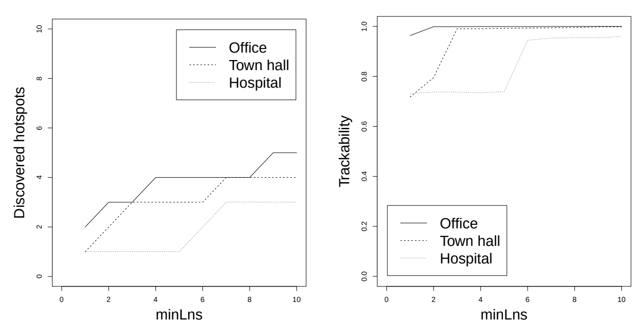
Fig. 12. Effect of *minLns* varying the amount of simulated persons at the university environment

for *minPts* $> 0.6|T|$, indicating that distinct clusters are found within these environments. In case of the town hall, the two discovered hotspots for *minPts* $= |T|$ suffice for a nearly complete tracking of all trajectories which is a good result. In case of the hospital more than 75% can be tracked using at least two nodes, and 94% with 5 nodes. However, for the office environment, the discovered hotspots show an overall impractical trackability, where only 84% of the total length of sub-trajectories can be tracked using 2 nodes.

D. Trajectory-Based Approach

Node placement based on representative trajectories is a very promising method, due to focusing on trajectories which already represent the most significant pedestrian flows. However, the clustering results of TRACLUS are critical referred to the required parameters ϵ and *minLns*. The ϵ-neighborhood of a line segment is forming an ellipsoid, rather than a circle as in case of DBSCAN when using points. Hence, we cannot set $\epsilon = \frac{1}{4}r_m$ like in the previous approach and have to evaluate the quality of discovered hotspots according to our constraints made in Section IV-C3. Again, we vary both parameters with $1 \leq \epsilon \leq r_m$ and $1 \leq minLns \leq |T|$ and investigate both, the amount of discovered hotspots and the corresponding trackability for different simulations and environments. Our investigations indicate, that for $\epsilon \approx \frac{1}{8}r_m$, and *minLns* ≤ 10 adequate clusters are found. For *minLns* > 10 more and more line segments are declared as noise and important clusters are destroyed. The effect of *minLns* for 10, 100, and 200 simulated persons choosing 5 destinations at the university building with $\epsilon = \frac{1}{8}r_m$ is shown in Figure 12. In contrast to the previous approaches, the amount of simulations highly influence the results w.r.t *minLns*, as depicted in Figure12(a). In case of 10x5 simulations, a maximum amount of 20 hotspots is discovered with *minLns* $= 3$ leading to a nearly complete coverage of all sub-trajectories. For *minLns* > 3 the amount of discovered hotspots decreases, due to an increased number of line segments which are marked as noise leading to destroyed clusters. In contrast, more simulations augment the density of trajectories and, hence, less but bigger clusters are found for a small value of *minLns*. When increasing *minLns*, more but smaller clusters are discovered leading to more representative trajectories. The higher the amount of simulations, the higher the density of start and endpoints of representative trajectories and the more can be grouped by DBSCAN and represented by one single hotspot, as observed for both dashed lines in Figure 12. The best result (optimal trade-off between a minimum amount of required nodes and a maximum trackability) for 100x5 simulations is obtained with *minLns* $= 4$ tracking 97.9% of the complete length of trajectories using 17 nodes. In case of 200x5 simulations, 15 hotspots are discovered with *minLns* $= 6$ tracking 98.3% in best case.

Again, the results for the other environments using 200x5 simulations with $\epsilon = \frac{1}{8}r_m$ are depicted in Figure 13. Like before, the amount of discovered hotspots increases with higher values for *minLns*, as depicted in Figure 13(a). Obviously, the effect of *minLns* is not as high as in case of a huge environment, due to a higher density of trajectories in small buildings. Only 3, 3, and 2 hotspots suffice to track 99.8% (*minLns* $= 2$), 99.3% (*minLns* $= 6$), and 94.4% (*minLns* $= 6$) at the office, town hall, and hospital, respectively. Again, these results announce the best trade-off between a minimum amount of required nodes and a maximum trackability.



(a) Hotspot discovery (b) Trackable trajectories

Fig. 13. Effect of *minLns* using 200 persons selecting 5 destinations at the other environments

E. Tracking Quality

The goal is to evaluate the quality of efficient Wi-Fi based monitoring with respect to the trackability when using the discovered hotspots. For this purpose, we use the most naive method and select randomly the same amount of nodes as we get by the performed approach on the used environment. The node positions are selected from the complete set of possible positions which are all white pixels of the used plant layout in our case. Note, that the optimal node placement is also included in this set of possible positions. In order to reach always the optimal solution, every possible k-node permutation of all n possible positions have to be checked according to the trackability which requires $\binom{n}{k}$ operations. This is too complex in case of huge bitmaps, though. Therefore, we use an approximate solution. We repeat the selection of node positions 100,000 times and take the best result with respect to its trackability. Beside the fact, that the selection is still random, we assume that this method approximates the optimal solution of node placement, due to the high amount of repetitions. Thus, we take the result as a quality measure for the discovered hotspots when using optimal input parameters.

Table I depicts the results for each environment using the proposed approach with the denoted parameters on 200x5 simulated paths. In order to evaluate and compare the quality of discovered hotspots, the last two rows show the trackability using the stated amount of nodes at the discovered hotspots, or at the best randomly selected positions. Considering the trackability, it is shown that there is only a marginal difference of up to 0.137 between the best random selection of positions and the discovered hotspots for each approach, and for all environments. This indicates that the proposed methods return a reliable setting of node positions for an efficient Wi-Fi based monitoring system. However, the density-based approach never returns a setting of hotspots showing a higher or equal trackability than the corresponding random selection. Therefore, we conclude that clustering of points representing only the direction changes of human paths is not an optimal placement strategy w.r.t the proposed quality measure, because it never reaches the optimal solution within the given settings. This is caused by the fact, that other characteristics, e.g. the trajectories' length, are not considered by this approach and, hence, longer trajectories may not be covered by any node leading to a decreased trackability.

We overcome this problem using the trajectory-based approach which considers the complete course of important sub-trajectories. With this method, we obtain the best results compared to our quality measure. Besides from the town hall, the trajectory-based approach always reaches a higher trackability than the corresponding best random selection and, thus, node placement based on representative trajectories indicates a sophisticated and quite optimal solution. According to these results, Figure 14 illustrates the 15 discovered hotspots as little circles within the university building. It is shown, that the depicted hotspots apparently mark strategic places inside the building, due to the fact, that each entrance door and every edge of principle hallways are observed by at least on node. This indicates a reasonable setting for Wi-Fi monitor nodes. However, the required parameters, such as *minLns*, and ϵ of the TRACLUS algorithm highly influence these results and, thus, they have to be selected carefully w.r.t to the given settings as it is shown in the previous section.

In contrast, the grid-based approach requires only one certain threshold ε which allows to find an optimal trade-off between a minimum amount of nodes and a maximal trackability. As depicted in Table I, adequate results have been found for $0.7 \leq \varepsilon \leq 0.83$. Compared to the best random selection, the grid-based approach performed better in the hospital, equal in the town hall, but worse in the office, and the university building. However, it still shows better results than the density-based approach and requires less input parameters. Therefore, this simple approach can be seen as a first systematic method to find good places for monitor nodes, but it cannot replace a more advanced solution like the trajectory-based approach.

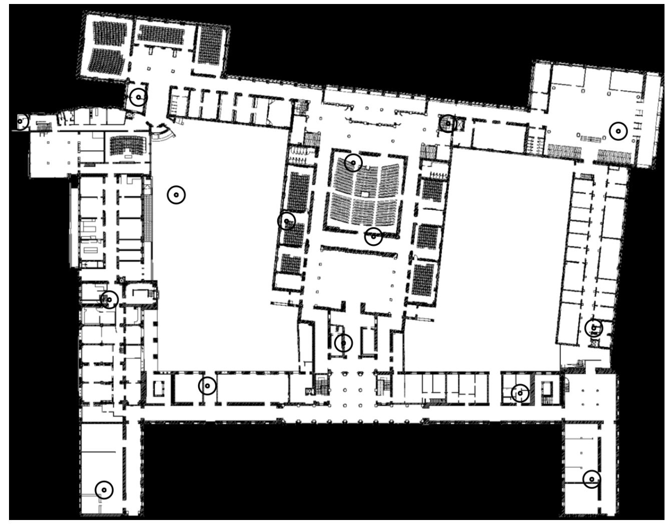


Fig. 14. Little circles denote the hotspots in the university building discovered by the trajectory-based approach.

VI. Conclusion and Future Work

In this paper, we have addressed the need for an adequate placement strategy of monitor nodes inside buildings for an efficient Wi-Fi based trajectory monitoring system. In order to find the best places for monitor nodes, which can be seen as a variant of the well-studied and NP-hard art gallery problem, three novel approaches for discovering hotspots have been presented. The proposed methods all work on bitmaps of typical plant layouts and simulated trajectories generated according to a modified pathway mobility model. While the first approach only considers local information from a heatmap, the last two perform density-based clustering on different characteristics of trajectories.

The evaluation of the proposed approaches on different environments and settings leads to the conclusion, that the presented methods allow for a systematic discovery of adequate node positions compared to a completely random selection of nodes. However, the density-based approach using DBSCAN on characteristic points never reaches a trackability higher than the proposed quality measure. In contrast, the grid-based approach is very simple, shows better results, and requires less input parameters. It can be seen as an easy and systematic method to find quite adequate node positions, but it does not always return the best solution for the tested environments w.r.t our quality measure. As a more sophisticated solution, the trajectory-based approach performs best and returns a quite optimal solution for most environments. However, it is more complex and requires more input parameters which have to be determined carefully, due to their high influence on the results.

In summary, we conclude that the presented approaches can reliably support the decision process for an adequate node placement of monitor nodes within a certain building. The evaluation has shown, that even our most simple approach leads to a better trackability using less monitors than an arbitrary

Environment	University			Office			Town Hall			Hospital		
Approach	Grid	Density	Trajectory	Grid	Density	Trajectory	Grid	Density	Trajectory	Grid	Density	Trajectory
Parameters	ε=0.73	$\epsilon\frac{1}{4}r_m$ *minPts*=25	$\epsilon = \frac{1}{8}r_m$ *minLns*=6	ε=0.83	$\epsilon = \frac{1}{4}r_m$ *minPts*=25	$\epsilon = \frac{1}{8}r_m$ *minLns*=2	ε=0.73	$\epsilon = \frac{1}{4}r_m$ *minPts*=200	$\epsilon = \frac{1}{8}r_m$ *minLns*=3	ε=0.70	$\epsilon = \frac{1}{4}r_m$ *minPts*=30	$\epsilon = \frac{1}{8}r_m$ *minLns*=6
# Hotspots	13	15	15	4	2	3	3	2	3	3	5	2
Trackability (Approach)	0.929	0.912	0.982	0.997	0.842	0.999	1.000	0.997	0.991	0.952	0.941	0.944
Trackability (Random)	0.979	0.981	0.981	1.000	0.979	0.998	1.000	0.999	1.000	0.947	0.991	0.928

TABLE I. AMOUNT OF NODES WITH REACHED TRACKABILITY USING HOTSPOTS COMPARED TO THE BEST RANDOM SELECTION OF NODE POSITIONS

installation of nodes, as often performed in related work. In principle, the findings of this paper are not only usable for Wi-Fi monitor nodes. The proposed approaches are also adoptable for any kind of wireless node placement, such as Bluetooth beacons or other sensor nodes. However, we focus on Wi-Fi, because we plan to realize a real world deployment of Wi-Fi monitor nodes for pedestrian flow monitoring on the basis of the discovered hotspots for future work. Furthermore, we plan to find a suitable heuristic to determine the optimal parameters for the clustering approaches in a given environment, and we want to enhance our evaluation focusing on other quality measures within bigger indoor environments.

REFERENCES

[1] T. Ahmed, T. B. Pedersen, and H. Lu. Capturing hotspots for constrained indoor movement. In *Proceedings of the 21st International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 472–475. ACM, 2013.

[2] F. Bai and A. Helmy. A survey of mobility models. *Wireless Adhoc Networks. University of Southern California, USA*, 206, 2004.

[3] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa. Signals from the crowd: uncovering social relationships through smartphone probes. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 265–276. ACM, 2013.

[4] B. Bonné, A. Barzan, P. Quax, and W. Lamotte. Wifipi: Involuntary tracking of visitors at mass events. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE, 2013.

[5] A. Bottino and A. Laurentini. A nearly optimal algorithm for covering the interior of an art gallery. *Pattern recognition*, 44(5):1048–1056, 2011.

[6] Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 900–911. IEEE, 2011.

[7] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[9] Y. Fukuzaki, M. Mochizuki, K. Murao, and N. Nishio. A pedestrian flow analysis system using wi-fi packet sensors to a real environment. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 721–730. ACM, 2014.

[10] H. González-Baños. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 232–240. ACM, 2001.

[11] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, 64:296–307, 2014.

[12] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin. Simultaneous optimization of sensor placements and balanced schedules. *Automatic Control, IEEE Transactions on*, pages 2390–2405, 2011.

[13] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, pages 235–284, 2008.

[14] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[15] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 281–294. ACM, 2012.

[16] L. Schauer, M. Werner, and P. Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *Proceedings of 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. EAI, 2014.

[17] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 85–98. ACM, 2009.

[18] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.

[19] K. Zheng, Y. Zheng, X. Xie, and X. Zhou. Reducing uncertainty of low-sampling-rate trajectories. In *Data Engineering (ICDE), 28th International Conference on*, pages 1144–1155. IEEE, 2012.