IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 December 2022, accepted 24 January 2023, date of publication 6 February 2023, date of current version 13 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242946

RESEARCH ARTICLE

Cellular Sidelink Enabled Decentralized Pedestrian Sensing

STEFAN SCHUHBÄCK[1,2], (Graduate Student Member, IEEE), LARS WISCHHOF[2], (Senior Member, IEEE), AND JÖRG OTT[1], (Member, IEEE)

[1]Department of Computer Engineering, TUM School of Computation, Information and Technology, Technical University of Munich, 85748 Garching, Germany
[2]Department of Computer Science and Mathematics, Munich University of Applied Sciences, 80335 Munich, Germany

Corresponding author: Stefan Schuhbäck (stefan.schuhbaeck@hm.edu)

This work was supported in part by the FORWIN of the Munich University of Applied Sciences; in part by the Graduate Center for Doctoral Studies in Informatics and its Applications (CeDoSIA) of Technical University of Munich (TUM) Graduate School at the Technical University of Munich, Germany; and in part by the Federal Ministry of Education and Research of Germany in the framework of roVer under Project 13FH669IX6.

ABSTRACT Pedestrian-based mobile sensing enables a large number of urban-centric use cases in the areas of intelligent mobility, smart city, and crowd management. With increasing standardization in Vehicle-to-Everything (V2X) communication to increase localized environmental awareness, i.e. cooperative perception (CP), a technological basis is already heavily discussed. Work in this area is usually directed toward road safety use cases. However, the same technologies could also be applied to pedestrian-centric applications in urban areas. Use cases like spatiotemporal density maps of pedestrians for public transportation optimization or urban route planning are such examples. This paper introduces an opportunistic decentralized mobile crowd sensing (MCS) approach where arbitrary measurement quantities are collected, aggregated, and disseminated in decentralized pedestrian measurement maps (DPMM). The sensing, dissemination, and aggregation are driven by mobile devices, without the need for centralized aggregation and dissemination infrastructure. By utilizing cellular sidelink communication (i.e. via the PC5 interface in 5G/6G systems) and node-local aggregation, the perception of the environment can be directly shared with neighboring nodes. The described DPMM approach is evaluated using CrowNet, an open-source simulation framework based on OMNeT++/INET by employing several detailed simulation studies: first, using a synthetic measurement quantity with a linear change rate behavior and second, a real use case concerning decentralized pedestrian density measurements. The results indicate that DPMM can provide spatiotemporal maps of the local area with a high level of detail and low delay – close to the optimum achievable in a specific mobility situation – while only requiring a moderate amount of cellular bandwidth.

INDEX TERMS Cellular sidelink, CrowNet, mobile crowd sensing, network simulation, pedestrian communication, pedestrian simulation, Vadere.

I. INTRODUCTION

The shared perception of the environment is a key aspect for many use cases in smart cities and urban-centric intelligent transportation systems (ITS). Possible use cases are: tracking live environmental quantities such as temperature, air quality, CO_2 concentration, humidity, noise pollution, or spatiotemporal pedestrian density maps for safety and ITS-related use cases. These density maps can provide real-time route recommendations [1] for public transportation and enable dynamic crowd management to increase safety, reduce crowded areas, and possibly decrease egress times. Another use case for pedestrian density maps, related to the spread of Covid, is that the density map allows pedestrians to circumvent high-density areas, thus reducing their exposure risk imposed by dense crowds [2].

With the ubiquity of devices such as smartphones, wearables, or portable IoT devices, pedestrians are major

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi.

contributors to mobile crowd sensing (MCS) applications. With use cases requiring shared or collective perception, pedestrians are both producers and consumers of the collected and aggregated data. MCS systems, however, primarily build centralized services for sensing campaign management, data aggregation, and dissemination. This might lead to privacy concerns if for instance pedestrian location data is used to create pedestrian density or flow maps.

Moving aggregation and dissemination-related tasks into the pedestrian devices and utilizing direct broad- and groupcast communication provided by 4G/5G/6G cellular sidelink, shared perception can be created without the need for centralized services to handle privacy-relevant data.

Based on previous work on disseminating pedestrian density measures [2], in this paper a generalized architecture for measurement maps, to collect, aggregate, and disseminate arbitrary spatiotemporal quantities is introduced. This enables the sharing of locally generated perception in a privacy-preserving fashion.

The main contributions of this paper are:

1) A generalized architecture to create, aggregate, and disseminate spatiotemporal quantities using broadcast communication technologies without the need for centralized services.
2) The application of this generalized architecture to the use case of pedestrian density measurements, using active position beacons and heuristic selection algorithms.
3) The publicly available open-source simulation framework CrowNet to conduct simulation studies incorporating the measurement map architecture.
4) A detailed simulation study analyzing four different scenarios: First a synthetic linear sensor model to analyze the effect of mobility and measurement quantity change rates. Furthermore, three scenarios utilizing the pedestrian density map are investigated in a stationary flow, a static scenario with impulse response characteristics, and a real-world example concerning a public transportation setting.

The article is structured as follows: First, in Section II, related work concerning the collection, aggregation, and dissemination of sensor values and the relation to other mobile crowd sensing (MCS) approaches is given. Next, Section III introduces the generalized architecture with the three tasks *Sensing*, *Dissemination*, and *Aggregation* followed by Section IV applying the architecture to pedestrians density sensing. Section V introduces metrics for evaluating the performance, the simulation model, and gives an overview of the CrowNet simulation framework and parameter settings used for the simulation study in Section VI. A conclusion and future work are given in Section VII. Tbl. 1 lists used abbreviations.

TABLE 1. Abbreviation table.

Abbreviation	Description
AOI	Area of Interest
CAM	Cooperative Awareness Message
CBR	Channel Busy Ratio
CP	Cooperative Perception
D2D	Device-to-Device
DPMM	Decentralized Pedestrian Measurement Map
FIFO	First In First Out
FSFT	First Seen First Transmitted
ITS	Intelligent Transportation System
LDM	Local Dynamic Map
MCS	Mobile Crowd Sensing
MEC	Mobile Edge Computing
MSCE	Mean Squared Cell Error
MSME	Mean Squared Map Error
NT	Neighborhood Table
TTL	Time-to-Live
V2X	Vehicle-to-Everything
VAM	Vulnerable Awareness Message
VRU	Vulnerbale Road Users
YMF	Youngest Measurement First
yDist	Youngest Measurement First plus Distance

II. RELATED WORK

In the following, related work to our proposed decentralized measurement map is presented: mobile crowd sensing, collective perception based on the cellular sidelink and concepts for measuring pedestrian densities.

A. MOBILE CROWD SENSING

The notion of building a shared perception, among mobile agents, is part of many research fields: The field of Mobile Crowd Sensing (MCS) looks at systems where local data collection or sensing is done by crowds utilizing mobile devices in conjunction with data aggregation in the cloud [3]. For MCS, multiple surveys classify existing MCS approaches along multiple dimensions: Capponi et al. [4] used four layers (application, data, communication, and sensing) to describe MCS system architecture. Similarly, Phuttharak et al. [5] used four dimensions (task, participation, data collection, and processing) to group MCS systems. We focus here on MCS systems without user interactions, i.e. opportunistic sensing [6] where the device performs the task autonomously. Such use cases are environmental monitoring like noise pollution mapping [7], intelligent transportation systems (ITS) applications for bus arrival time estimation [8], or pedestrian density estimation by applying moving sensors [9].

A majority of MCS applications rely on centralized structures for data aggregation, post-processing, and analysis, leaving only sensor data collection to the decentral components [4]. This entails that possible fine-grained data must be transmitted to centralized services, allowing the potential collection of privacy-relevant data combined with large communication loads. Further, if the aggregated data is again needed directly at the origin, it has to be queried from the backend system. Decentralized approaches where the aggregation and dissemination of the crowd-sensed measurement are less prevalent. Benedetto et al. [10] for instance propose to use early cellular side link capabilities like LTE-direct to transmit arbitrary multi-frame data in the LTE-direct discovery resource allocated slots. This broadcast communication allows direct sharing with neighboring

nodes. Costa et al. [11] describe a system for querying trajectories of pedestrians that are stored locally for privacy reasons. Applying edge and fog technology in the MCS context [12], [13] decreases network-induced delays but still leads to centralization of application data as the applied edge application is controlled by the same entity.

B. CELLULAR SIDELINK: COLLECTIVE PERCEPTION

Newer cellular networks have a dedicated link, i.e. Sidelink, for direct device-to-device (D2D) communication over the PC5 interface for safety-relevant and proximity-based applications. Work on cellular Sidelink is related to the vehicle-to-everything (V2X) communication paradigm that is used in automated vehicles as they rely on environmental perception shared through wireless communication [14]. In ETSI standardization, environmental and situational awareness in the V2X context is separated into cooperative awareness (CA) and collective perception (CP). The former, CA, is part of the cooperate awareness basic service [15], where V2X nodes exchange information, using cooperative awareness messages (CAMs) about their states such as heading, speed, and location. In the same context, vulnerable road user (VRU) awareness messages (VAM) [16] are used to communicate the position and heading of VRU, i.e. pedestrians or cyclists. The objects described in the CAM and VAM messages are collected in each V2X node in a local dynamic map (LDM) [17], which thus contains the current perception of that node, created through shared knowledge.

However, the LDM only contains information generated by the participating nodes. With collective perception (CP) [18] nodes can share the knowledge they collected in their local environment model with other neighboring nodes to increase shared perception [19]. CP allows therefore an increased shared perception compared to the LDM structure. Work related to CP in V2X is for instance about redundancy mitigation measures [20] to reduce the channel busy ratio (CBR), without reducing the collective awareness provided by CP. Willecke et al. [21] investigate the effect of rules for the inclusion policy for VRU objects in collective perception messages. Similar to MCS approaches, the use of edge or mobile edge computing (MEC) [22] is considered to improve the use cases of CP as well as the overall dissemination of the shared environment model.

The shared collective perception generated in V2X use cases closely resembles our proposed decentralized pedestrian measurement map. However, pedestrians are only considered VRUs, and therefore included in the shared environment model when they participate in road traffic in a safety-relevant way [16]. If pedestrians are in a pedestrian zone they would not create VAM messages and are thus not included in the shared environment model.

C. PEDESTRIAN DENSITY MEASUREMENT

Measuring or estimating pedestrians densities in urban areas can be achieved in multiple ways: Image and video-based approaches as discussed and surveyed in [23], [24], and [25] infer pedestrian densities and flows from single images up to multiple camera angles by applying, for instance, deep learning algorithms. Most of these systems utilize stationary-mounted camera systems. However, Tokuda et al. [9] introduce the usage of images from moving sensors (i.e. cars) for pedestrian counting. Common disadvantages of image-based approaches are the need to install cameras in the area of interest (AOI) beforehand as well as the privacy implication entailed with vast centralized image processing.

Non-vision-based approaches like [26] use positional data from cellular network operators to extract pedestrian density data. Wirtz et. al [27] describe a participatory sensing system for crowd monitoring based on GPS data collected through smartphone applications. Other approaches investigate for instance Wi-Fi probing behavior [28] to count pedestrians in a privacy-preserving manner. These approaches, however, focus on the collection and aggregation and do not discuss the dissemination of the results to increase the shared perception.

In [29], a theoretical system for pedestrians and cyclists sensing (AMSense) is proposed, which describes the usage of vehicular sensors and V2X communication to capture the spatiotemporal properties of pedestrians and cyclists. The system relies on vehicular sensors and has architectural elements where the localized aggregation is applied with support from edge and cloud data processing.

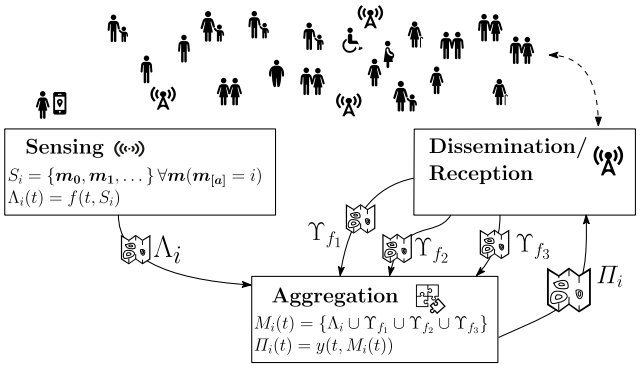


FIGURE 1. Architecture of DPMM. It shows the interaction between the three tasks *Sensing*, *Aggregation*, and *Dissemination/Reception*. The Local map (LM) Λ_i, generated in the *Sensing* task, is used together with received foreign aggregated maps (FAMs) Υ_*, provided by the *Dissemination/Reception* task, in the *Aggregation* task to create the local aggregated map (LAM) Π_i. This map is then broadcasted to neighboring nodes.

III. DECENTRALIZED MEASUREMENT MAPS

In the following, we define a decentralized measurement map as a collection of arbitrary spatiotemporal data that is created by sharing the data between mobile and or stationary agents. With the measurement map, agents can share their perception of the environment with neighboring agents, to create and improve their understanding of the environment. For this, each agent translates its geospatial measurement of the environment into a measurement map. The environment is discretized into squared, non-overlapping cells. If an agent has information about one of these cells, it assigns a

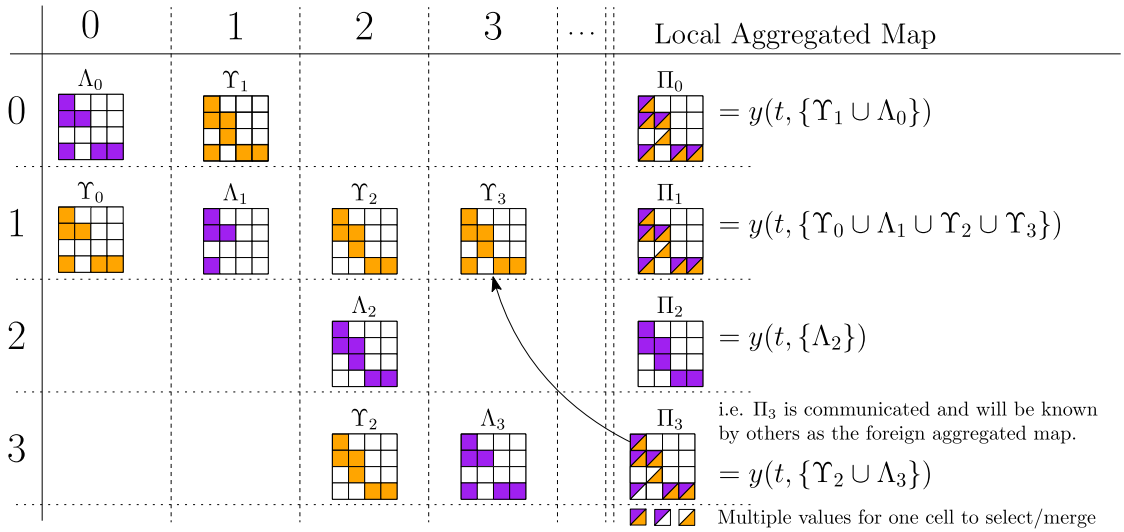


FIGURE 2. Data driven view of the measurement map build process. Each row represents the view of the respective node. It contains received foreign maps Υ and sensed local maps Λ (on the main diagonal). These maps provide the basis for the calculation of the local aggregated map Π. For example: Node 1 receives foreign maps Υ from Node 0, 2, and 3. Together with the own measurements, Λ_1, the local aggregated map Π_1 can be created. Node 2 does not receive any data from other nodes, thus only the local measurements, Λ_2, can be used to calculate the aggregate map Π_2.

measurement value as well as a set of metadata elements to this cell. These maps are then shared with neighboring agents using broadcast communication. The receiving agents use these foreign maps to improve and increase their perception of the environment before communicating their updated measurement maps. Fig. 1 shows the three tasks and their relationship on an architectural level. The three tasks are *Sensing*, *Dissemination*, and *Aggregation*. Fig. 2 gives a data-driven view of the measurement map creation process. It shows where data is located and how it is shared among participating agents. With the three tasks, three versions of the measurement map are introduced to describe the creation and ownership nature of the data. The three map types are: the local measurement (LM) map Λ, the foreign aggregated measurement (FAM) maps Υ, and the local aggregated measurement (LAM) map Π.

In the *Sensing* task, each agent creates a local perception of its surrounding area by collecting sensor readings. This might be a temperature sensor, collecting temperature and humidity readings, an air quality sensor tracking the CO_2 level, or some other setup that for instance allows the counting of pedestrians by utilizing active position beacons from which the number of pedestrians for a certain area can be derived.

The data obtained from these local sensors are processed in the *Sensing* task, such that the collected data is assigned to cells of the local measurement (LM) map denoted as Λ, see Fig. 1 upper right and Fig. 2 on the main diagonal. This map contains a set of measurements that are based on the local sensor readings only.

The *Dissemination* task consists of two subtasks, namely the reception of foreign aggregated measurement (FAM) maps Υ and the dissemination of the local aggregated measurement (LAM) map Π. The received foreign maps Υ will be used in the *Aggregation* task to create the LAM map Π of the current node. There are multiple ways how this aggregation can be conducted. Aggregation algorithms will heavily depend on the type of measurement, i.e. temperature or the number of pedestrians, as well as available contextual metadata.

The second subtask addresses the actual dissemination of the generated LAM map Π. The map is divided, based on measurement cells, into smaller chunks to be sent as one-hop broadcast packets to neighboring nodes. Each packet consists of a header and 1-to-N cell measurements, with N depending on the maximum transmission unit (MTU) and the cell measurement size. Packets are independent of each other to ensure that the loss of one does not invalidate other packets. The task also collects received map packets from foreign agents into foreign aggregated measurement (FAM) maps Υ, which are the counterpart to the LAM map, see Fig. 2 (arrow between Υ_3 and Π_3). The dissemination task ensures that each FAM map is updated when a new map packet from the corresponding foreign agent arrives and provides access to these maps when the aggregation task uses them to generate the LAM map.

Next, a formal definition of key structures for the measurement maps is given, followed by a definition of the input and output structures for each task. A list of used symbols is given in Tbl. 2.

A. MEASUREMENT MAP

We describe each measurement map Λ, Υ, and Π as a set of measurements. Each measurement $\boldsymbol{m}$ is defined as a pair in (1a), containing the measurement value ρ and a set of additional metadata elements m^{meta} shown in (1b).

$$\boldsymbol{m} = (\rho, m^{meta}), \text{ with} \tag{1a}$$

TABLE 2. Symbol table.

Symbol	Description
Λ	Local Measurement (LM) Map
Υ	Foreign Aggregated Measurement (FAM) Map
Π	Local Aggregated Measurement (LAM) Map
S_i	Set of measurements created by a sensor model
A_M	Set of agent identifiers contributing measurements to a set of measurements M
C_M	Set of cells referenced in set of measurements M
I_C	Order set of cells indicating transmission order
E_f	List of measurements contained in one map packet received from foreign agent f
τ_{NT}	Time-To-Live NT data
τ_{Map}	Time-To-Live Map data
k_c	Cell knowledge ratio
$\hat{C}_{ped}$	Mean pedestrian count
$\hat{o}$	Occupancy ratio
$\Delta\hat{t}^O/\Delta\hat{t}^E$	Occupancy/Empty mean interval length
$f(\cdot)$	Sensor model function
$g(\cdot)$	FAM update function
$y(\cdot)$	Aggregation function to create Π from Λ and Υ
$\boldsymbol{m}$	Measurement object.
$\boldsymbol{m}_{[\rho]}$	Measurement value of $\boldsymbol{m}$ assigned to cell c.
$\boldsymbol{m}_{[c]}$	Cell to which the measurement $\boldsymbol{m}$ belongs to.
$\boldsymbol{m}_{[t]}$	Measurement $\boldsymbol{m}$ creation time.
$\boldsymbol{m}_{[a]}$	Agent identifier $a \in A$ which created the measurement $\boldsymbol{m}$.
$\boldsymbol{m}_{[d]}$	Distance between cell and generating agent $\boldsymbol{m}_{[a]}$ at time t.
$\boldsymbol{b}$	Beacon object.
$\boldsymbol{b}_{[a]}$	Creating node identifier.
$\boldsymbol{b}_{[\vec{x}]}$	Node postion at creation time.
$\boldsymbol{b}_{[t]}$	Beacon creation time.
$\boldsymbol{b}_{[c]}$	Cell of node at beacon creation time.

$$m^{meta} = \{c, t, a, d\} \tag{1b}$$
$$M = \{\boldsymbol{m_0}, \boldsymbol{m_1}, \boldsymbol{m_2} \dots\} \tag{1c}$$

The metadata set m^{meta} in (1b), contains associated data points to give contextual meaning to the measurement. This includes, for instance, the cell identifier c which marks the geospatial point the measurement is associated with. Furthermore, the set contains the creation time t of the measurements as well as the identifier a of the agent, which created the measurement, and the distance d between the respective cell and the creating agent at the creation time t.

To simplify the access of the measurement tuple entries defined in (1), the following notation is used: Given a measurement $\boldsymbol{m}$, access to the measurement value and metadata is written as $\boldsymbol{m}_{[\square]}$ where $\square$ is substituted with the respective selector. For example, $\boldsymbol{m}_{[t]}$ is the creation time of $\boldsymbol{m}$, $\boldsymbol{m}_{[c]}$ is the cell linked to the measurement. Likewise, $\boldsymbol{m}_{[\rho]}$ will be the measurement value of the measurement $\boldsymbol{m}$, see Tbl. 2 for a full list.

Next, we define the age function $\Delta_{\boldsymbol{m}}$ in (2), which calculates the age of a measurement based on the creation time $\boldsymbol{m}_{[t]}$ of the measurement.

$$\Delta_{\boldsymbol{m}} : \mathbb{R} \mapsto \mathbb{R}; (t) \mapsto \Delta_{\boldsymbol{m}}(t) = t - \boldsymbol{m}_{[t]} \tag{2}$$

Lastly, given a set of measurements M, as defined in (1c), we define common sets that will be used throughout the definition of the measurement maps. The set of agents A_M in (3a) contains all agent identifiers, which contribute measurements into the set M. Let C_M in (3b) further describe the set of all cells for which at least one measurement is present in M.

$$A_M = \left\{ \boldsymbol{m}_{[a]} \middle| \boldsymbol{m} \in M \right\} \tag{3a}$$
$$C_M = \left\{ \boldsymbol{m}_{[c]} \middle| \boldsymbol{m} \in M \right\} \tag{3b}$$

B. SENSING

In the sensing task, each agent i uses some sensor model to create a set of measurements S_i for cells the agent can perceive, see (4).

$$S_i = \{\boldsymbol{m_0}, \boldsymbol{m_1}, \dots\} \text{ with } \forall \boldsymbol{m}(\boldsymbol{m}_{[a]} = i) \tag{4}$$

At some time interval, the agent uses the set of measurements to create the LM map $\Lambda_i(t)$ for the current time t. We assume that the sensor model provides a temporal discretization, which does not have to coincide with the update cycle used for the LM map. The agent applies function $f(\cdot)$ which takes the current time t as well as the set of measurements S_i to create the LM map $\Lambda_i(t)$, see (5a).

$$\Lambda_i(t) = f(t, S_i) \text{ with} \tag{5a}$$
$$\forall \boldsymbol{m}, \boldsymbol{n} \in \Lambda_i(t)(\boldsymbol{m}_{[c]} \neq \boldsymbol{n}_{[c]}) \text{ and} \tag{5b}$$
$$\forall \boldsymbol{m} \in \Lambda_i(t)(\boldsymbol{m}_{[a]} = i) \tag{5c}$$

The definition of $f(\cdot)$ depends on the sensor model. The only requirements that the function f must meet are that for each cell at most one measurement exists (5b) and that each measurement must belong to the current agent i (5c).

C. DISSEMINATION

The dissemination task consists of two subtasks: transmission and reception.

1) TRANSMISSION SUBTASK

The measurement map is communicated to agents through local one-hop broadcast utilizing sidelink communication. The measurement map is divided into multiple, independent packets at the application layer to ensure that the loss of one does not lead to the loss of the whole map. The size of the packets is limited by the maximum transmission unit of the used link.

The fragmentation strategy of the measurement map relies on multiple factors such as the type of measurement value, the inherent dynamic nature, the size of the area of interest, and the available resources. The strategy will influence the order and the transmission frequency with which each cell measurement is communicated. A detailed analysis and comparison of these strategies will be left for future work.

In this paper, we assume a fragmentation strategy with a constant order and an identical inter-transmission interval for all cells. A simple algorithm would be a ring buffer of all cells from which cells are selected for the next packet. To mitigate line or row scanning effects, introduced by a simple row/column order of cells, we use a first seen first

transmitted (FSFT) approach based on the first in first out (FIFO) principle. At the start of the application, we assume an empty ordered set of cells $I_C = \emptyset$. After the first aggregation step, the node has access to the initial LAM map $\Pi_i(t)$. The FSFT strategy then uses the creation time of the cell measurements to create a transmission order. The ordered set I_C is then extended every time the LAM map is updated such that new cells, which were not seen before, are appended at the end. This creates a fixed transmission order for each node based on local knowledge alone (see Algorithm 1).

Algorithm 1 FSFT Strategy for Cell Transmission Order

Input: $\Pi_i(t)$, $I_C(t_{-1})$ previous ordered set of cells
Output: $I_C(t)$
1: M := sort $\Pi_i(t)$ by creation time $\boldsymbol{m}_{[t]}$
2: $I_C(t) := I_C(t_{-1})$
3: **for** $\boldsymbol{m}$ in M **do**
4: **if** $\boldsymbol{m}_{[c]} \notin I_C(t)$ **then**
5: append cell $\boldsymbol{m}_{[c]}$ to $I_C(t)$
6: **end if**
7: **end for**
8: **return** $I_C(t)$

This order is kept throughout the lifetime of the measurement map. It only defines the sequence in which each cell is checked for valid measurements. If a measurement reached the time to live τ_{Map}, the measurement is removed from the LAM map but the corresponding cell stays in the transmission order set $I_C(t)$ and will be skipped if no valid measurement exists.

2) RECEPTION SUBTASK

On the receiving side, the reception subtask handles the reception of map packets E_f and their assembly into the FAM map Υ_f, where the suffix f denotes a foreign agent from which the packet originated. The update of the FAM map $\Upsilon_f(t)$ is defined by the update function $g(\cdot)$ as shown in (6a), ensuring that after the update, at most one measurement for each cell exists (6b) and that all measurements are created by the agent f (6c).

$$\Upsilon_f(t) = g(t, \Upsilon_f(t_{-1}), E_f) \text{ with} \tag{6a}$$
$$\forall \boldsymbol{m}, \boldsymbol{n} \in \Upsilon_f(t)(\boldsymbol{m}_{[c]} \neq \boldsymbol{n}_{[c]}) \text{ and} \tag{6b}$$
$$\forall \boldsymbol{m} \in \Upsilon_f(t)(\boldsymbol{m}_{[a]} = f) \tag{6c}$$

The function uses the previous state of FAM map $\Upsilon_f(t_{-1})$ as well as the currently received packet E_f to create the FAM map $\Upsilon_f(t)$. If no previous state of the FAM map exists, let $\Upsilon_f(t_{-1}) = \emptyset$ be the empty set in (6a).

As an update function $g(\cdot)$, we use a filter as described in Algorithm 2: First, it ensures that all measurements in $\Upsilon_f(t_{-1})$ and E_f are not older than the time to live τ_{Map}. Next, for each measurement in the filtered set E, it is tested if a measurement for this cell already exists, see line 5 in Algorithm 2. If so, only the most recent measurement, i.e. the one with the newer creation time stamp, is selected. If no measurement associated with this cell exists, the measurement is appended to the FAM map.

Algorithm 2 Update Algorithm $g(\cdot)$ to Create $\Upsilon_f(t)$ on Map Packet E_f Reception

Input: $\Upsilon_f(t_{-1})$, E_f
Output: $\Upsilon_f(t)$
1: Y := filter $\Upsilon_f(t_{-1})$ by age: $\forall \boldsymbol{m}(\Delta_{\boldsymbol{m}}(t) < \tau_{Map})$
2: E := filter E_f by age: $\forall \boldsymbol{m}(\Delta_{\boldsymbol{m}}(t) < \tau_{Map})$
3: C_Y := set of cells of measurements in Y (see (3b))
4: **for** $\boldsymbol{m}$ in E **do**
5: **if** $\boldsymbol{m}_{[c]} \in C_Y$ **then**
6: $\boldsymbol{n}$:= measurement in Y for cell $\boldsymbol{m}_{[c]}$
7: **if** $\boldsymbol{m}_{[t]} > \boldsymbol{n}_{[t]}$ **then**
8: replace $\boldsymbol{n}$ with $\boldsymbol{m}$ in Y
9: **end if**
10: **else**
11: append measurement $\boldsymbol{m}$ to Y
12: **end if**
13: **end for**
14: **return** Y (updated set of measurements $\Upsilon_f(t)$ at time t)

D. AGGREGATION

With the definition of the LM map $\Lambda_i(t)$ in (5) and the FAM map $\Upsilon_f(t)$ in (6) for foreign agents, the current agent i has the set of measurements to its disposal to create the LAM map $\Pi_i(t)$.

$$M_i(t) = \left\{ \bigcup_{f}^{F_i \subset A} \Upsilon_f \cup \Lambda_i \right\} \text{ with } f, i \in A \text{ and } i \notin F_i \subset A \tag{7}$$

The suffixes f and i in (7) are node identifiers, where A is the set of all agent identifiers and F_i, a real subset of A, containing the foreign agent identifiers from which the current agent i has a FAM map Υ_f.

The set $M_i(t)$ from (7) is the input to the aggregation function $y(\cdot)$ which generates the LAM map $\Pi_i(t)$ of the agent i.

$$\Pi_i(t) = y(t, M_i(t)) \text{ with} \tag{8a}$$
$$\forall \boldsymbol{m}, \boldsymbol{n} \in \Pi_i(t)(\boldsymbol{m}_{[c]} \neq \boldsymbol{n}_{[c]}) \text{ and} \tag{8b}$$
$$\forall \boldsymbol{m} \in \Pi_i(t)(\boldsymbol{m}_{[a]} = i) \tag{8c}$$

The aggregation function $y(\cdot)$ computes the new estimates for each cell while ensuring that for each cell only one measurement exists, see (8b). Furthermore, the function must ensure that all measurements returned belong to agent i, even if foreign measurements are used as input (8c). This is done to ensure that no foreign identifiers are communicated to preserve privacy.

In the following, we use a simple selection-based approach as the aggregation function. This class of heuristics selects one measurement based on the assigned rank provided by

a rank function defined by $r(\boldsymbol{m}, t)$ in (9). This function maps a real-valued rank between $0 \leq r(\boldsymbol{m}, t) \leq 1$ to each measurement $\boldsymbol{m} \in M_{i,j}$ at time t.

$$r : (M_{i,j}, t) \mapsto \mathbb{R}; (\boldsymbol{m}, t) \mapsto r(\boldsymbol{m}, t) \tag{9}$$

The set $M_{i,j} \subseteq M_i(t)$ describes a subset of $M_i(t)$, containing all measurements in $M_i(t)$ concerning cell j. Let further be $C_{M_i(t)}$ the set of cells present in the set of measurements $M_i(t)$ as defined in (3b).

$$M_{i,j} = \left\{ \boldsymbol{m} \middle| \boldsymbol{m} \in M_i(t) \wedge \boldsymbol{m}_{[c]} = j \right\} \tag{10}$$

The selection is then applied to each cell present in $C_{M_i(t)}$ as described in Algorithm 3. First, an empty LAM map $\Pi_i(t)$ is created. Next, for each cell $j \in C_{M_i(t)}$, the measurement with the smallest rank, provided by the specified rank function, is selected. For the selected measurement $\boldsymbol{m'}$, the agent identifier is set to the current agent. Lastly, the measurement $\boldsymbol{m'}$ is appended to the LAM map $\Pi_i(t)$. This algorithm ensures the constraints in (8b) and (8c), such that the LAM map Π_i only contains at most one measurement for each cell with the correct agent identifier.

Algorithm 3 Selection Based Aggregation Algorithm

Input: $M_i(t)$, rank function $r(\boldsymbol{m}, t)$
Output: $\Pi_t(t)$

```
Π_t(t) := ∅
for j ∈ C_{M_i(t)} do
    /* select m with smallest rank for cell j. */
    m' := argmin_{m∈M_{i,j}} r(m, t)
    m'_[a] = i /* update agent identifier. */
    append m' to Π_t(t)
end for
return Π_t(t)
```

One simple example of a selection-based aggregation heuristic is an age-based approach, which utilizes the age of the measurement as a rank function. This simple heuristic was introduced in [2] under the name youngest measurement first (YMF). The rank is given by the age in (11), such that the newest measurement will be selected. We use this aggregation method as a baseline due to its simplicity.

$$r_{ymf}(\boldsymbol{m}, t) = \Delta_{\boldsymbol{m}}(t) \tag{11}$$

IV. DECENTRALIZED PEDESTRIAN DENSITY MAPS

This section applies the generic concept to a typical application of crowd sensing, namely the sensing of pedestrian densities in the local area. The aim is to create a decentralized, shared perception of the pedestrian density in the close to medium range by aggregating position beacons, which are communicated by all agents. The measurement quantity saved in the measurement map value $\boldsymbol{m}_{[\rho]}$ is the number of pedestrians in each cell. The beacon and the measurement map use the same agent identifiers such that beacons and map messages from one agent can be matched. However, each node only communicates its identifier and never rebroadcasts any foreign received identifiers.

A. SENSING OF PEDESTRIAN DENSITY

The pedestrian density is calculated based on the number of pedestrians in a given cell. The location and dimension of each cell are shared knowledge. To derive the number of pedestrians, a beacon sensing approach is used, where each agent actively communicates its location via recurring position beacons. Therefore, each agent i receives a set of beacons B_i (12a), which are described as a tuple shown in (12b). Based on the set of received beacons, the agent creates a neighborhood table (NT) that contains one entry of the form $\boldsymbol{b}$ for each agent.

$$B_i = \{\boldsymbol{b_0}, \boldsymbol{b_1}, \boldsymbol{b_2}, \dots\} \tag{12a}$$

$$\boldsymbol{b} = (a, \vec{x}, t, c) \tag{12b}$$

The selection notation $\boldsymbol{b}_{[\square]}$ for the NT entry values is similar to the definition of the measurement tuple described in (1). The entry encodes the sending agent's identifier $\boldsymbol{b}_{[a]}$, position $\boldsymbol{b}_{[\vec{x}]}$, and beacon creation time $\boldsymbol{b}_{[t]}$. Additionally, the cell identifier $\boldsymbol{b}_{[c]}$ based on the agent position is included for convenience. Note that the age function in (2) is also applicable to NT entries $\Delta_{\boldsymbol{b}}(t) = t - \boldsymbol{b}_{[t]}$. Similar to the set of agent identifiers contained in a set of measurements in (3a), let $\mathcal{A}_{B_i}$ in (13) be the set of agent identifiers from which at least one beacon was received.

$$\mathcal{A}_{B_i} = \left\{ \boldsymbol{b}_{[a]} \middle| \boldsymbol{b} \in B_i \right\} \tag{13}$$

The NT contains only the newest location of each agent, thus only the newest beacon is included in NT. (14) creates the subset $B'_i \subseteq B_i$ of beacons that adheres to this condition. It selects the newest beacon from one source a (i.e. $\{\boldsymbol{b} | \boldsymbol{b} \in B_i \wedge \boldsymbol{b}_{[a]} = a\}$), and combines the result in a union over all beacon sources $\mathcal{A}_{B_i}$, as defined in (13).

$$B'_i = \left\{ \bigcup_{a}^{\mathcal{A}_{B_i}} \underset{\{\boldsymbol{b} | \boldsymbol{b} \in B_i \wedge \boldsymbol{b}_{[a]} = a\}}{\operatorname{argmin}} (\Delta_{\boldsymbol{b}}(t)) \right\} \tag{14}$$

Lastly, the set B'_i is filtered, such that only beacons that are younger than some time to live τ_{NT} are part of the neighborhood table, (15).

$$NT_i(t) = \left\{ \boldsymbol{b} \middle| \boldsymbol{b} \in B'_i \wedge \Delta_{\boldsymbol{b}}(t) \leq \tau_{NT} \right\} \tag{15}$$

The base LM map Λ_i is then generated by calculating the number of agents for one cell $\rho_{c,i}$ by summing up all beacons that originated in that cell.

$$\rho_{c,i} = \sum_{\boldsymbol{b}}^{NT_i(t)} u(c, \boldsymbol{b}) \text{ with } u(c, \boldsymbol{b}) = \begin{cases} 1 & \text{if, } \boldsymbol{b}_{[c]} = c \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The measurement value $\rho_{c,i}$ is then combined with the necessary metadata information into a measurement value $\boldsymbol{m}$ for cell c, generated by agent i at time t. Compare with (1).

$$\boldsymbol{m} = (\rho_{c,i}, \{c, t, a = i, d\}) \tag{17}$$

Applying (16) and (17) for all cells present in the NT, agent i generates the base LM map $\Lambda_i(t)$ for the current point in time. However, to update the LM map over time, the map and NT must be updated together each time a new beacon arrives. If one cell is occupied by one agent and this agent moves to a different cell, the number of pedestrians calculated in (16) only reports the new location of the agent but does not include the knowledge that the previous cell is now empty. The process of updating the NT and LM map is described in Algorithm 4 and Algorithm 5.

Every time a beacon is received, the NT will be checked if an entry reached the time to live τ_{NT}, see Algorithm 4. If an entry reached τ_{NT}, it is removed from the neighborhood table. In addition, the corresponding measurement $\boldsymbol{m} \in \Lambda_i(t)$, of the LM map is updated by decrementing the count and updating the metadata, i.e the generation time $\boldsymbol{m}_{[t]}$.

Algorithm 4 Check Time to Live in Neighborhood Table (NT)

Input: $NT, \Lambda_i, t, \tau_{NT}$
1: **for** $\boldsymbol{b} \in NT$ **do**
2: **if** $\Delta_{\boldsymbol{b}}(t) > \tau_{NT}$ **then**
3: retrieve cell measurement $\boldsymbol{m}$ for NT entry $\boldsymbol{b}$
4: decrement count and update metadata in $\boldsymbol{m}$
5: delete entry $\boldsymbol{b}$ from NT
6: **end if**
7: **end for**

Next, the received beacon is processed as shown in Algorithm 5: If the beacon is already out of date, it will not be processed. Otherwise, the existing NT entry $\boldsymbol{b'}$ and map entry $\boldsymbol{m'}$ concerning the cell from where the beacon originated are selected if present. If no entry is found, the beacon is the first entry from this source. The corresponding cell measurement from the LM map is selected (or created if not existing) and incremented. It is possible that the LM map measurement already exists because the cell might be occupied by other agents. In case $\boldsymbol{b'}$ exists, it is tested if the source agent moved between cells. If not, no increment is needed and only the time stamp for the LM map measurement is updated. If a cell change occurred, the old cell measurement is decremented and the new cell measurement is incremented accordingly. Lastly, the state of the neighborhood table is updated if needed.

To summarize, the LM map $\Lambda_i(t)$ is continuously updated by using received position beacons and the neighborhood table to create a localized perception of the pedestrian density by counting beacons.

B. DISSEMINATION

The pedestrian density map does not need a specialized dissemination process. The generic algorithms described in Sec. III are sufficient to receive and disseminate pedestrian density map packets.

Algorithm 5 Update Local Measurement (LM) Map Λ_i

Input: $\Lambda_i, NT, t, beacon$
Output: Λ_i
1: $\boldsymbol{b}$:= create new NT entry from beacon
2: **if** $\Delta_{\boldsymbol{b}}(t) > \tau_{NT}$ **then**
3: /*TTL reach; Beacon dropped, no update needed */
4: **return** Λ_i
5: **end if**
6: $\boldsymbol{b'}$:= access old NT entry if present
7: $\boldsymbol{m'}$:= access cell measurement for $\boldsymbol{b'}_{[c]}$ if present
8: **if** $\boldsymbol{b'}$ = null **then**
9: create new cell measurement $\boldsymbol{m}$ for cell
10: increment count and update metadata of $\boldsymbol{m}$
11: **else if** $\boldsymbol{b}_{[c]} = \boldsymbol{b'}_{[c]}$ **then**
12: access cell measurement $\boldsymbol{m}$ of cell
13: only update metadata of $\boldsymbol{m}$, agent stayed in cell
14: **else**
15: /*agent moved to new cell*/
16: access old cell measurement $\boldsymbol{m'}$
17: decrement count and update metadata of $\boldsymbol{m'}$
18: access/create new cell measurement $\boldsymbol{m}$
19: increment count and update metadata of $\boldsymbol{m}$
20: **end if**
21: update NT
22: **return** Λ_i

C. AGGREGATION

In addition to the YMF algorithm described in the previous chapter, we introduce the adapted version, Youngest measurement first plus distance (yDist), to better cope with the beacon-based sensing approach.

The rationale behind the YMF heuristic is that newer data more closely describes the current state of the cells. However, preliminary results in stationary scenarios indicated that for cells at the edge of the coverage, the number of pedestrians can be underestimated. A possible reason is that own, beacon-based, measurements are favored since these values are updated more frequently. The update frequency of the LM map is closely linked to the beacon frequency as well as the number of agents perceived. Each beacon will update the time stamp of the corresponding measurement as described in Algorithm 5. On the other hand, updates generated by FAM packets are linked to the lower map packet frequency, producing a lower updated frequency for these measurements and consequently a larger average age.

In the case of the YMF aggregation, this will lead to situations where local measurements are preferred because they have a smaller average age, compared to the foreign measurements due to the difference in their respective reception frequency.

If the respective cell is close by, this has no effect because the own sensing and received aggregated results show similar counts. However, if a cell is located at the edge of the reception range of the current node, parts of the cell's area may be outside of the reception range, leading to blind spots.

If there are agents in these blind spots, the LM map will underestimate the number of agents.

This kind of degradation of the measurement accuracy is specific to the beacon sensing since the measured value is derived from the number of received packets. For other measurement values such as temperatures where the value is part of the message, lost messages just lead to missing values and not wrongfully low values. These wrongfully low values cannot be distinguished from correct low agent counts. Note that lost measurement map packets do not introduce this problem. Measurement map packets do encode the value as the content of the message. Lost map packets just lead to missing values and not erroneous low values.

The error of underestimation at the edge of the reception range is a problem with the data coming from the LM map Λ. Other agents, which are closer to the cells at the edge of the current agent's reception range do not have this problem. However, by the time the foreign agent communicates these cell measurements through a FAM map Υ, these measurements are older and will be dismissed by the YMF heuristic. To mitigate this, the distance between the measuring node and the cell measured is introduced in the decision process. The yDist algorithm is defined by the rank function $r_{yDist}(\boldsymbol{m}, t)$ which assigns each measurement $\boldsymbol{m}$ a rank based on the combination of distance and age.

$$r_{yDist}(\boldsymbol{m}, t) = \alpha\Gamma(\boldsymbol{m}, t) + (1-\alpha)\Psi(\boldsymbol{m}, t) \tag{18}$$

The rank function (18) is defined as the convex combination of an age rank $\Gamma(\boldsymbol{m}, t)$ with a distance rank $\Psi(\boldsymbol{m}, t)$. The resulting rank is thus the weighted average of both ranks with the weights α and $(1-\alpha)$, with $0 \leq \alpha \leq 1$.

The distance rank $\Psi(\boldsymbol{m}, t)$ defined in (20) uses the relative source-cell distance of each measurement as a rank. The distance rank should only affect measurements at the edge of the measurement distance. Thus, the cutoff distance parameter $D > 0$ is introduced in (19) which will give all measurements smaller or equal to the cut of distance the same rank.

$$I_D(d) = \begin{cases} D & \text{if } d \leq D, \\ d & \text{otherwise.} \end{cases} \tag{19}$$

The distance rank $\Psi(\boldsymbol{m}, t)$ thus uses the step function $I_D(d)$ to calculate the rank in (20).

$$\Psi(\boldsymbol{m}, t) = \frac{I_D(\boldsymbol{m}_{[d]})}{\sum_{\boldsymbol{m} \in M_{i,j}} I_D(\boldsymbol{m}_{[d]})} \tag{20}$$

For the age rank $\Gamma(\boldsymbol{m}, t)$ in (22), we compare the normalized age difference between the current age and the minimum age (21) of each measurement for cell j.

$$t_{min} = \min_{\boldsymbol{m} \in M_{i,j}(t)} (\Delta_{\boldsymbol{m}}(t)) \tag{21}$$

We use the age difference rather than the age because otherwise, the relative difference between two ranks would change over time, even if no new measurements are introduced. In combination with the distance rank, this would lead to the possibility that the ranking changes over time without introducing new measurements. To mitigate this, the age difference is used, which will stay constant as time progresses.

$$\Gamma(\boldsymbol{m}, t) = \begin{cases} \frac{\Delta_{\boldsymbol{m}}(t) - t_{min}}{Z_\Gamma(M_{i,j})} & \text{if } Z_\Gamma(M_{i,j}) > 0 \\ 1.0 & \text{otherwise.} \end{cases} \tag{22}$$

with

$$Z_\Gamma(M_{i,j}) = \sum_{\boldsymbol{m} \in M_{i,j}} (\Delta_{\boldsymbol{m}}(t) - t_{min}) \tag{23}$$

Note that (23) might reach zero if there is only one measurement or when all measurements have the same age. In this case, the rank is set to $\Gamma(\boldsymbol{m}, t) = 1.0$ to get the same behavior as in the distance rank.

The aggregation function for the yDist heuristic uses the selection-based aggregation Algorithm 3, combined with the rank function $r_{yDist}(t, \boldsymbol{m})$ shown in (18).

V. METRICS AND SIMULATION MODEL

In this section, identified metrics are introduced, which will be used in the simulation study to evaluate our model. Secondly, the simulation model and base simulation parameters are introduced, in conjunction with the used simulation framework CrowNet.

A. METRICS

1) ERROR METRICS

We choose the mean squared cell error (MSCE) (24a) to evaluate the accuracy of the pedestrian map. The MSCE is based on the squared error of each cell measurement $(Y_{i,j}(t) - \hat{Y}_j(t))^2$ for all agents i which have a measurement $Y_{i,j}(t)$ for cell j. $\hat{Y}_j(t)$ denotes the ground truth for the cell j. The term $Y_{i,j}(t)$ is the measurement value $\boldsymbol{m}_{[\rho]}$ which is part of the LAM map $\Pi_i(t)$ of the agent i for the cell j, see (24b). The number of agents $|A_t|$ will change over time due to new agents entering or leaving the area of interest (AOI). To get an error metric for some AOI, the MSCE of the containing cells are combined in the mean squared map error (MSME) as shown in (24c).

$$MSCE_j(t) = \frac{1}{|A_t|} \sum_{i}^{A_t} \left(Y_{i,j}(t) - \hat{Y}_j(t) \right)^2 \tag{24a}$$

$$Y_{i,j}(t) = \boldsymbol{m}_{[\rho]} \text{ with } \boldsymbol{m} \in \Pi_i(t) \wedge \boldsymbol{m}_{[c]} = j \tag{24b}$$

$$MSME(t) = \frac{1}{|C|} \sum_{j}^{C} \left(MSCE_j(t) \right) \tag{24c}$$

2) CELL KNOWLEDGE RATIO

The number of agents that have knowledge about a given cell, denoted by $|A_t|$ in (24a), does not have to be the number of all agents currently in the area of interest. If agents in the area of interest build disconnected clusters, one cluster does not know about the other. The error metric as defined

in (24) will not show that only a portion of the agents provided measurements for the metric. This problem of missing values can be addressed in different ways. In the case of pedestrian density maps, missing values can be replaced with a count of zero pedestrians. The rationale behind this approach is, that the sensing of the pedestrian is based on active beacons, meaning that there is no way to count zero pedestrians. The absence of measurements can therefore be assumed to be a measurement of zero pedestrians. Missing values should be removed, when such an imputation approach is not justifiable. In this case, the error metrics should be associated with a coverage ratio k_C, which indicates how many agents participated in the error calculation. For the measurement maps, this ratio is defined as the number of agents which have a measurement for the given cell $|A_t|$ divided by the number of all agents $|A'_t|$ currently in the area of interest (25). For pedestrian density maps, the ratio will be $k_C = 1.0$ due to the use of the imputation approach.

$$k_C = \frac{|A_t|}{|A'_t|} \tag{25}$$

3) MEAN PEDESTRIAN COUNT

For the analysis of the pedestrian density map, the pedestrian count over time $\hat{C}_{ped}(t)$ (26) of the mean density map is used to test how accurately the number of pedestrians in the AOI can be estimated using density maps. The number of pedestrians $\hat{C}_{ped}(t)$ is calculated by first summing up the pedestrian count each agent i has in their LAM map $\Pi_i(t)$ at the current time, i.e. the total number of pedestrians seen by the current agent. The agent-based map counts are then aggregated over all agents currently present at time t, denoted as A_t, to form the mean pedestrian count.

$$\hat{C}_{ped}(t) = \frac{1}{|A_t|} \sum_{i}^{A_t} \sum_{\boldsymbol{m}}^{\Pi_i(t)} \boldsymbol{m}_{[\rho]} \tag{26}$$

4) CELL OCCUPATION RATIO AND INTERVAL DISTRIBUTION

The measurement accuracy, and therefore the error, depends on the capability of the system to capture the dynamics of the underlying measurement quantity. Parameters with influence are the available communication resources and the packet dissemination frequency.

Besides these standard parameters, the mobility characteristics of the sensing nodes (i.e. chosen path, speed, and interaction with the environment) as well as the range of the used sensors play a role in the reachable accuracy. Due to the free movement of nodes through the scenario, cells will go in and out of coverage. That means the set of cells for which the measurement quantity can be sensed will change over time based on the mobility pattern. We call the time a cell is in the sensor range of at least one agent the occupancy time of that cell.

During this time, the node can sense the measurement quantity of the cell. At times when the cell is empty, i.e. out of reach of the sensing range of all nodes, no measurements

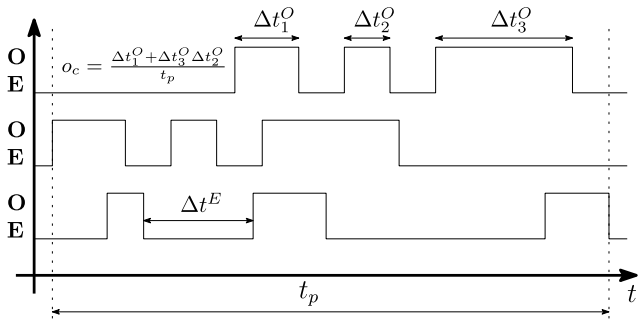


FIGURE 3. Example of three cells showing exemplary occupation pattern and occupancy ratio as defined in (27a). The plot illustrates intervals of occupation Δt^O, i.e. at least one node is able to sense values of this cell, and intervals Δt^E, where no node can sense the measurement quantity of the cell.

can be taken. During these time intervals, the system cannot track changes in the measurement quantity. Already taken measurements will age over time. Errors that arise from this condition cannot be eliminated by increasing the dissemination interval or providing more communication resources. This condition is highly dependent on mobility patterns and the overall topography, which in conjunction will create some interval distribution (i.e. length and frequency of occupied and empty intervals). To make this circumstance more tangible, we introduce the cell occupation ratio and the mean cell occupied/empty interval length. Fig. 3 shows an exemplary occupancy pattern of three cells over time. Each cell has two possible states: empty **E**, no measurements possible, and occupied **O**, a cell can be measured. The occupation ratio o_c (27a) is the time a single cell c is occupied divided by the analysis period. The occupancy time is the sum of all intervals N^O in the analysis period t_p. See Fig. 3 for an example. The average occupation ratio for the AOI and analysis period, containing cells C, is given by (27b).

$$o_c = \frac{t_c^o}{t_p} = \frac{\sum_i^{N^O} \Delta t_i^O}{t_p} \tag{27a}$$

$$\hat{o} = \frac{1}{|C|} \sum_{c}^{C} o_c \tag{27b}$$

The mean occupied and empty interval for an area of interest is given by creating the mean over all occupied intervals N^O, respectively all empty intervals N^E over all cells.

$$\hat{\Delta t^O} = \frac{1}{|N^O|} \sum_{i}^{N^O} \Delta t_i^O \tag{28a}$$

$$\hat{\Delta t^E} = \frac{1}{|N^E|} \sum_{i}^{N^E} \Delta t_i^E \tag{28b}$$

To summarize, the accuracy of the system depends on (1) the underlying dynamics of the measurement quantity, (2) the map and sensing frequency, (3) the available communication resources, and (4) the cell's occupancy interval length and distribution over time.

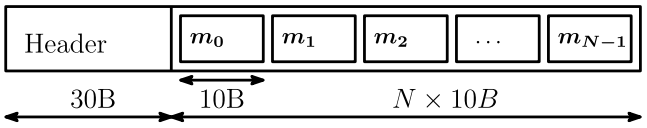


FIGURE 4. Measurement map packet structure.

B. SIMULATION MODEL

In this section, we describe the used simulation model, framework, implementation details of the measurement and density map, as well as the default simulation parameters used in the communication and mobility domain.

1) CrowNet SIMULATION FRAMEWORK

The simulation study is conducted with the CrowNet simulation framework. The framework allows the simulation of packet-level communication based on the OMNeT++ ecosystem (INET [30], Simu5G [31], Artery [32], VEINS [33]), combined with detailed microscopic mobility simulators for pedestrians and vehicles. Existing pedestrian mobility models such as the striping model in Sumo [34] are useful but do not accurately represent pedestrian mobility patterns in open spaces such as pedestrian zones. This is due to the graph-based routing in the Sumo striping model. For this, mobility models, such as the Optimal Steps Model (OSM), where pedestrians can move freely in the 2D space are better suited. The effect of different detailed pedestrian mobility models was previously analyzed in [35]. The OSM utilizes floor fields that represent target attraction to model routes, as well as obstacles and agent repulsion, to ensure topographic bounds and prevent overlapping pedestrians. For the superposition of these fields, each agent calculates the next optimal step by solving an optimization problem within their respective step circle. For an in-depth model description, the reader is referred to [36], which contains detailed descriptions of the used OSM implementation provided by the Vadere pedestrian dynamics simulator.

In the CrowNet framework, we previously introduced [37] a bi-directional coupling between the communication and mobility domain by utilizing the Traffic Control Interface (TraCI) [38]. This approach is based on the existing work of the frameworks VEINS [33] and Artery [32] to allow the simulation of coupled scenarios where mobility and communication domains affect each other.

In the used scenarios the communication domain does not interact with the mobility domain. Therefore, we use pre-recorded mobility traces, generated by Vadere, to increase the simulation performance. The traces are based on the BonnMotion trace format introduced in [39]. To allow the dynamic creation and removal of agents based on the Vadere traces, an adaption of the INET BonnMotion mobility module is necessary. We publish the open-source CrowNet framework, including all required adaptions to the INET modules and all simulations for this study on GitHub.[1]

[1] https://github.com/roVer-HM/crownet/tree/pub_ieeeAcces_DPMM

2) SIMULATION MODEL

The measurement and density map are implemented as map data structures, containing the cell measurements either directly created based on the sensing step, i.e. the LM map Λ, or based on received FAM maps Υ. The LAM map will be generated on demand based on the selected aggregation algorithm. The algorithm selection is based on the strategy design pattern and will be configured for each simulation. A similar approach is used in the Dissemination/Reception task, where the configured dissemination strategy selects cells for transmission.

For the packet structure, we assume the following setup shown in Fig. 4: The map header is 30 bytes long, contains a sequence number (2B), a node identifier (4B), the lower 32 bit of a 64-bit milliseconds timestamp as well as 18 bytes to encode the grid origin and cell size. The time stamp will wrap every 49.7 days which is long enough for 1-hop broadcast packets. With the timestamp, the sequence number, and node identifier, out-of-order and duplicate packets can be detected and removed. The payload shown in Fig. 4 consists of a list of 10-byte long blocks, encoding the measurement value for one cell measurement $\boldsymbol{m}$. It contains at least the measured value, creation time, and cell identifier. Similarly, the beacon packet (36 bytes) contains a sequence number, identifier, time stamp, and location information. The packet structures can be further tuned to reduce header and data size when taking into account the underlying measurement quantity and other contextual metadata elements. The analysis of this and further decentralized configuration capabilities of the measurement map system will be left for further work.

3) SIMULATION PARAMETERS

All simulations utilize the Mode 3 sidelink communication with the support of the base station. Our key use cases are based in the urban area, thus the assisted mode is chosen. The analyzed scenarios contain a single base station. Multiple base station scenarios as well as Mode 1 sidelink will be part of future work. Tbl. 3 shows the default parameters. We use a small base station with 25 resource blocks on the carrier frequency of 2.6 GHz. The propagation model provided by the Simu5G framework is configured as an urban microcell [40], which provides a heuristic path loss model. The channel model is further configured to take into account interference in up- and downlink as well as between D2D devices. For a detailed Simu5G model description, the reader is referred to [31], [41].

In Vadere, the Optimal Steps Model uses a truncated normal distribution to set the free flow speed of pedestrians based on a mean speed of 1.34 ms^{-1}. The pedestrian potential parameters ensure that there are no overlaps. The optimal steps model uses the discrete mode with a step circle resolution of four, which means that for each step four equidistant points on the step circle circumference are used from which the best next step is selected. A more detailed mobility modeling, like solving an optimization problem on

TABLE 3. Base simulation parameters.

Parameter	Value
Carrier frequency	2.6 Ghz
Resource Blocks (RB)	25
Path loss model	Urban microcell
UE height	1.5 m
Enb height	25 m
Building height	20 m
Shadowing	active
Interference	UL/DL/D2D
Dynamic loss	active (corr. distance 25 m)
AMC mode	D2D
Channel Quality Indicator D2D	7
RLC type	UM (unacknowledged mode)
RLC queue size	300 kB
MAC queue size	10 kB
Number eNB	1
τ_{Map}	S0: – / S1-2: 30.0 s / S3: 15.0 s
τ_{NT}	S0: – / S1-3: 5.0 s
Map transmission interval Δt_{Map}	100 ms..7500 s (*)
Beacon transm. interval Δt_{Beacon}	300 ms..1000 ms (*)
yDist$_{\alpha/D}$ parameters α / D	S0: – / S1-3: α: 0.0..1.0, D: 60 m..160 m (*)
Ped. mean free flow speed	1.34 ms^{-1}
Std. dev. of ped. free flow speed	0.26 ms^{-1}
Free flow speed maximum	2.5 ms^{-1}
Free flow speed minimum	0.5 ms^{-1}
Pedestrian radius	0.2 m
Ped. potential intimate space width	0.45
Ped. potential personal space width	1.2
Ped. intimate space factor	1.2
OSM step circle resolution	4

(*) Parameters varied in simulation studies. S0:3 mark different settings for each scenario. The full list of parameter settings for each scenario can be found on the publicly available repository[1].

the step circle disc is possible. However, the discrete option was chosen because the topographies in this paper are based on wide-open outdoor spaces and not highly accurate indoor scenarios like classrooms.

VI. SIMULATION STUDY

In this section, we analyze the decentralized measurement map in four distinct scenarios, where the first scenario (S0) uses the generic measurement map without any coupling between measurement quantity and measuring nodes. Scenarios S1-3 will apply the pedestrian density maps with beacon sensing where the measurement quantity (i.e. pedestrian count) depends on the number of nodes in the simulation.

Scenario (S0) will use an abstract measurement quantity with a linear change rate and a dual corridor topography with separated flows in opposite directions. In scenarios S1-3, the pedestrian density map with beacon sensing is used: Scenario S1 will use the same topography and mobility patterns as scenario S0 to analyze performance in a steady state flow. In Scenario S2, a stationary mobility pattern is used with the simultaneous removal of half of all agents to analyze the convergence time of the density map, void of any mobility-induced changes. Lastly, a dynamic scenario S3 concerning a subway entrance is used to conduct a parameter study for the yDist heuristic described above.

TABLE 4. Parameter variation for S0.

Δt_{Map}	change rate α			
	0.01 s^{-1}	0.10 s^{-1}	0.50 s^{-1}	1.00 s^{-1}
0.10 s	S0:0	S0:1	S0:2	S0:3
0.30 s	S0:4	S0:5	S0:6	S0:7
0.70 s	S0:8	S0:9	S0:10	S0:11
1.00 s	S0:12	S0:13	S0:14	S0:15
5.00 s	S0:16	S0:17	S0:18	S0:19
10.00 s	S0:20	S0:21	S0:22	S0:23
20.00 s	S0:24	S0:25	S0:26	S0:27
40.00 s	S0:28	S0:29	S0:30	S0:31
80.00 s	S0:32	S0:33	S0:34	S0:35
100.00 s	S0:36	S0:37	S0:38	S0:39
7500.00 s	S0:40	S0:41	S0:42	S0:43

A. ABSTRACT MEASUREMENT QUANTITY (S0)

The objective of this scenario is to see the effect of different inter-transmission intervals and measurement quantity dynamics (i.e. change rates) on the error quantity, given a fixed set of mobility patterns in a dual corridor topography. We use a linear model of the form shown in (29) for the measurement values, where the ground truth measurement $\hat{Y}_j(t)$ for a cell j over time is given by a linear equation with a random, cell-dependent parameter β_j and a system-wide change rate of α. We further assume that the *sensor range* of this abstract sensor model is limited to the cell the agent is currently part of.

$$\hat{Y}_j(t) = \beta_j + \alpha t \tag{29}$$

1) TOPOGRAPHY DESCRIPTION

The topography is displayed in Fig. 5. In the upper path, pedestrians are generated in a source area (green rectangle) on the right and move to the target area (orange rectangle) on the left. In the bottom path, the agents are generated on the left and move to the right. In the center of the scenario, one eNB is placed. The horizontal paths have a width of 5 m such that the paths only contain one row of square cells of the size of 5 m. Grey areas in Fig. 5 are obstacles that only affect the mobility pattern of pedestrians. Pedestrians are generated in each source with a constant inter-arrival time of 25 seconds, which translates to a system-wide pedestrian generation frequency of $\lambda_{X_{4.8}} = \frac{2\,ped}{25\,s} \cdot \frac{60\,s}{min} = 4.8\,\frac{ped}{min}$. The pedestrian flow is generated by Vadere using the optimal steps model with a free flow speed drawn from a truncated normal distribution of $\mathcal{N}(1.35\frac{m}{s}, 0.26\frac{m}{s})$.

2) MOBILITY DYNAMICS

For this scenario, we vary the change rate α and the map transmission interval Δt_{Map}, see Tbl. 4. Each variation is executed N=20 times, where the seed for both the

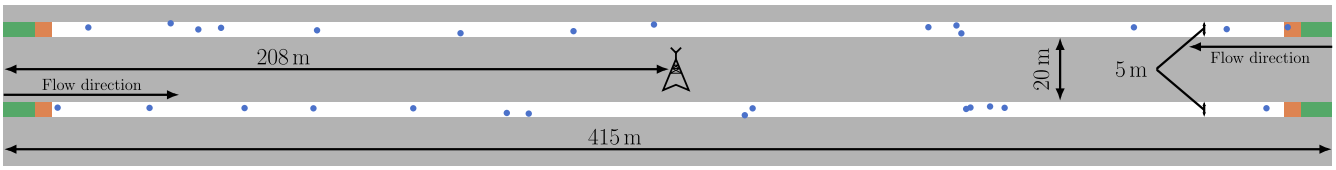


FIGURE 5. Topography of scenario S0/S1 with an area of 415m × 40 m. Pedestrians (blue circles, enlarged for figure) move from left to right on the lower corridor and from right to left on the upper corridor. The corridors are separated by obstacles (gray). Pedestrian that reach their target (orange rectangles) are removed from the simulation. Note that the obstacles only affect the mobility simulation.

communication and mobility domains are varied. In the mobility domain, the most prevalent effect of the seed variation will be different free-flow speeds and, derived from that, differences in the cell occupancy ratio and interval distributions. The mobility dynamics of the simulations are shown in Fig. 6. The mean cell occupancy is $\hat{o} = 0.1452$ overall seeds with a standard deviation of 0.015. Fig. 6a plots the histogram for each seed as well as the distribution over all seeds. It shows a similar distribution of the cell occupancy ratio overall seeds. Similarly, the mean empty interval length is depicted in Fig. 6b, depicting a heavy-tailed distribution toward longer interval lengths. The mean empty interval length, over all, seeds is $\hat{\Delta t^E} = 24.0035\,s$ with a standard deviation of 18.25 s. In comparison with the occupancy interval length of $\hat{\Delta t^O} = 4.6621\,s$ and the standard deviation of 1.5578 s, the empty interval length has a higher mean and a larger spread with a tailed distribution towards longer interval lengths.

3) SIMULATION RESULTS

The MSME metric for Scenario S0 is shown in Fig. 7. On the right, Fig. 7a, the MSME over time with the corresponding standard deviation over all seeds is shown. For small map transmission intervals, the error values are smaller, as expected due to higher sampling. The periodical change, most prevalent in S0:32 and S0:40, can be attributed to the inter-arrival time of pedestrians in the simulation. Looking at smaller map transmission intervals of $\Delta t_{map} \leq 5.0\,s$ in Fig. 7b, the errors are considerably smaller. Furthermore, these errors are in the same region as the pure mobility-induced error: If we assume an error-free communication with infinite resources and an infinite small map transmission interval, a cell can still only be measured if at least one agent is in the sensing range. As shown in Fig. Fig. 6c, a cell is empty on average for 24.0035 seconds in the scenario. Thus, the moment before an agent enters a cell, the currently known value of this cell is on average around 24 seconds old. The squared error e of this average age can be calculated using (29) as follows.

$$e = (\hat{Y}_j(t_1) - \hat{Y}_j(t_0))^2 \tag{30a}$$

$$e = (\alpha * (t_1 - t_2))^2 = (\alpha * \Delta t^E)^2 \tag{30b}$$

$$e = (0.01 * 24.0035)^2 = 0.057\,616\,801\,225 \tag{30c}$$

The red line in Fig. 7b depicts the mobility-induced error (30c) based on the average length of the empty interval. It illustrates that for small map transmission intervals, most of the error can be explained by the mobility pattern. Note further, that for transmission intervals smaller than $\Delta t_{Map} \leq 1.0\,s$, only small differences in the error metric are observed, see the zoomed area in Fig. 7b. This can also be seen in Fig. 7c, where the residual MSEME error ratio, compared to scenario S0:0 with a map transmission interval of $\Delta t_{Map} = 100\,ms$, is given. It demonstrated that even a ten-fold increase in the transmission interval to $\Delta t_{Map} = 1.0\,s$ (S0:12) does not increase the error metric considerably. Even an increase to 5.0 s only increases the error by 20%.

In Fig. 7c, the cell knowledge ratio k_C is shown. It indicates that each cell is known by all agents after some time. The oscillation in the cell knowledge ratio is explained by the creation of new agents every 25 seconds. These agents do not have any knowledge and must first receive information from neighboring nodes. With increasing map transmission intervals, the convergence time increases. If no communication is done, the knowledge ratio only depicts the own cells the agent enters, thus explaining the low knowledge ratio for simulation S0:40.

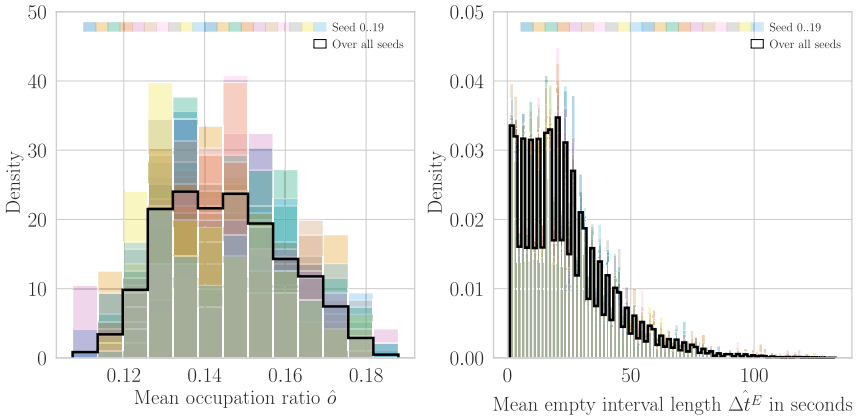
The results of the same structure were observed for other change rates, however with increased amplitude in the MSME metric. This is expected due to increased absolute measurement values. The results are therefore omitted here for brevity.

4) SUMMARY (S0)

The results demonstrate that the measurement map quality depends on the mobility pattern of the participating agents. If knowledge about the change rate of the measurement quantity and the mobility patterns is available (i.e. interval length of occupation), a lower bound estimate of the error can be given. Furthermore, estimates for the lower bounds for the transmission interval are possible, which will reduce the bandwidth requirements. Note further that the mobility patterns as well as the measurement rate of change can be derived locally, without the need for centralized services.

B. STEADY STATE SCENARIO (S1)

In this and the following scenarios, the pedestrian density map with beacon sensing is used, as defined in Sec. IV. In scenario S1 we evaluate if the pedestrian density map can model a stationary flow of pedestrians. The scenario establishes a stationary flow of pedestrians in opposite directions. It uses the same topography and mobility setup as described in Scenario S0, see Fig. 5. In addition to the



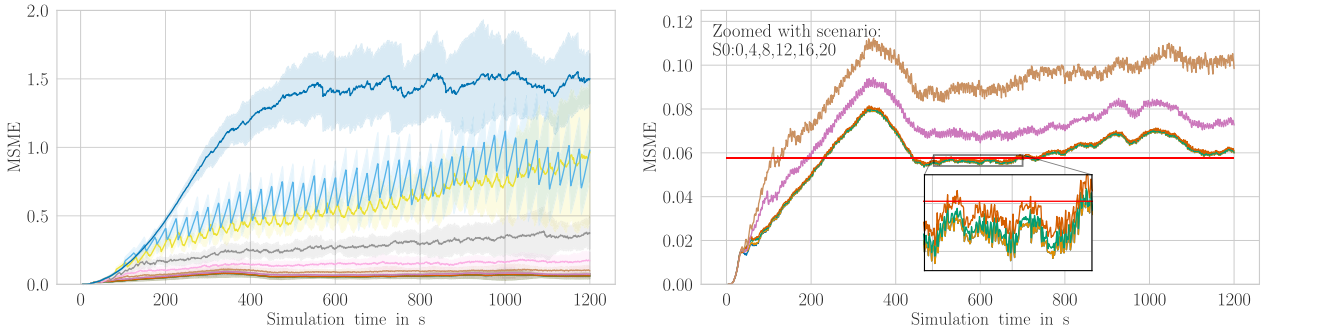
(a) Histogram of mean cell occupation ratio $\hat{c_c}$

(b) Histogram of mean interval length $\Delta\hat{t}^E$

Stat.	$\hat{o_c}$	$\Delta\hat{t}_E$	$\Delta\hat{t}_O$
Mean	0.1452	24.0035	4.6221
Std. Dev.	0.0150	18.2643	1.5578
Min	0.1073	1.0000	2.0000
25%	0.1333	11.0000	4.0000
50%	0.1440	20.0000	4.0000
75%	0.1560	32.0000	5.0000
Max	0.1880	133.0000	24.0000

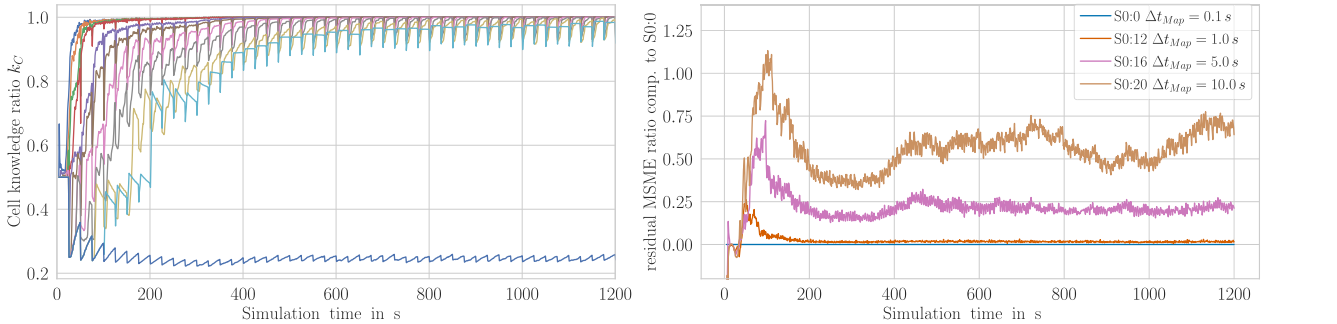
(c) Descriptve statistics.

FIGURE 6. Cell occupation ratio and interval distribution for N=20 seeds used for scenario S0.



(a) MSEM for all map transmission intervals showing a considerable error increase for larger transmission intervals.

(b) Zoom view of MSME. The red horizontal line depicts the MSME created soley due to the mean empty interval $\Delta\hat{t}^E = 24.0035\,s$



(c) Cell knowledge k_C ratio over time

(d) Residual error ratio comparing S0:12,16,20 to S0:0. Error of S0:20 is about 20% higher over time compared to S0:0.

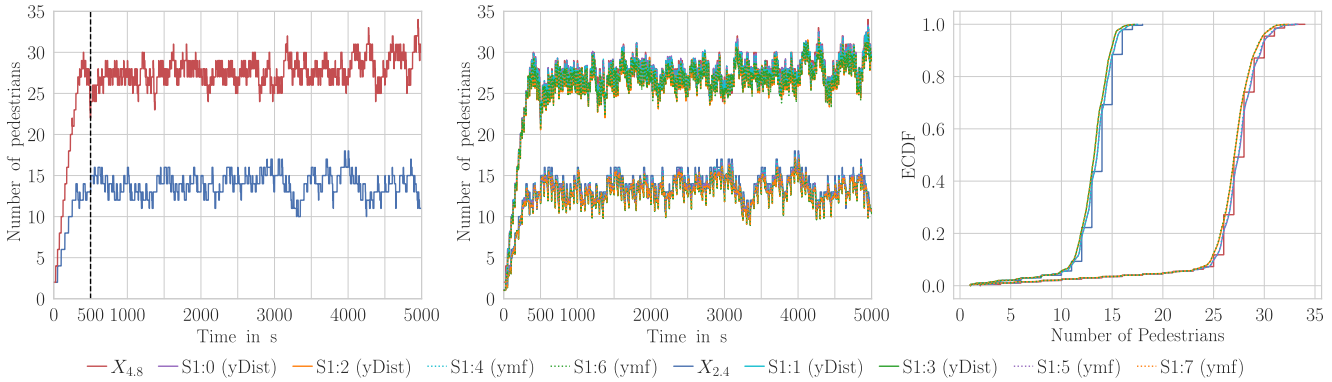
FIGURE 7. Scenario S0: Mean squared map error (MSME), cell occupation ratio and residual MSME ratio for change rate $\alpha = 0.01\,\mathrm{s}^{-1}$ and with map transmission interval from 0.1 s to 7500 s. The last scenario S0:40 depicts the situation without any communication and only local sensing knowledge.

inter-arrival time of 25 seconds in Scenario S0, a second inter-arrival time of 50 seconds, i.e. a pedestrian frequency of $\lambda_{X_{2.4}} = \frac{2\,ped}{50\,s} \cdot \frac{60\,s}{min} = 2.4\,\frac{ped}{min}$ is introduced. Furthermore, the simulation time is increased to 5000 seconds to ensure that the pedestrian flow is stationary over time.

Fig. 8a shows the time series $X_{2.4}$ and $X_{4.8}$ of the number of pedestrians in the simulation over time. After a ramp-up time of approximately 500 seconds, the number of pedestrians in the simulation visually follows a stationary process with an average number of pedestrians of $\bar{X_{2.4}} = 13.45$ and a sample variance of $s_{X_{2.4}} = 2.29$ for $X_{2.4}$ and $\bar{X_{4.8}} = 26.84$ and $s_{X_{4.8}} = 4.20$ for the time series $X_{4.8}$ respectively. Given the stationary criterion, the mean and variance do not change over time and thus can be used to describe the time series.

1) STATIONARITY TEST

To test the time series for stationarity, we apply the augmented Dickey-Fuller test (ADF). The null hypothesis is that the time series has a unit root and thus is not stationary. The alternative hypothesis is that the time series under test



(a) Ped. count over time for mobiliy traces (b) Ped. count over time for S1:0-7 (c) ECDF of mobiliy traces and S1:0-7

FIGURE 8. Scenario 1 (S1): Number of pedestrians in simulation over time. (a) Ground truth given by traces generated by Vadere for $X_{2.4}$ (blue) with pedestrian arrival frequency of $\lambda_{X_{2.4}} = 2.4\,\frac{ped}{min}$ and $X_{4.8}$ (red) with a pedestrian arrival frequency of $\lambda_{X_{4.8}} = 4.8\,\frac{ped}{min}$. Mean pedestrian count over time (b) shows close visual accordance with ground truth data. The small underestimation can be seen in the empirical cumulative density function (c), comparing all variations S1:0-7, which are shifted left, to their respective ground truth $X_{2.4}$ or $X_{4.8}$.

does not have a unit root and thus is stationary. Rows $X_{4.8}$ and $X_{2.4}$ in Tbl. 5 show the test score, p-value, and criticality criterion for the time series of the traces. The null hypothesis must be rejected, due to the small p-value as well as the test statistic value being smaller than the critical value. Given the stationary flow of pedestrians, the density map should be able to reproduce this process. We thus use the mobility traces of $X_{2.4}$ and $X_{4.8}$ in the simulation to test if the density maps can represent the stationary process. The mean pedestrian count $\hat{C}_{ped}$ (26), over time, is then compared to the respective ground truths $X_{2.4}$ and $X_{4.8}$.

2) PARAMETER VARIATION

For the evaluation of the stationarity, we apply a parameter study containing the parameters beacon and map interval, the pedestrian frequency, and the aggregation algorithm as shown in Tbl. 6. Each parameter variation is run with $N = 20$ different seeds concerning the communication simulation. Fig. 8b shows the time series of the number of pedestrians created by the density maps and Fig. 8c the empirical CDF of each variation and ground truth. Tbl. 5 shows the ADF test statistics and statistical moments for each variation sorted by pedestrian frequency. It shows that the mean pedestrian count $\hat{C}_{ped}$ of each variation S1:0-7 closely matches the ground truth data with a small underestimation of the mean pedestrian count. The p-value of the ADF test allows the rejection of the null hypothesis for a confidence level of 1%, meaning that the density map does create a stationary process. Further, the scenario shows that a higher frequency of beacon and map packets produces a smaller error in the mean pedestrian count, compare S1:0/2 or S1:4/6 for $\lambda_{X_{4.8}}$ in Fig. 8b and 8c. The difference is tested by applying a Mann-Whitney U test (H_0 assumes distributions are the same). Comparing the distributions of S1:0/2 $p_{S1:0/2} = 6.096 \times 10^{-28}$ and S1:4/6 $p_{S1:4/6} = 3.079 \times 10^{-26}$ the H_0 must be rejected for a confidence level of 5%, such that there is a difference in the central tendencies. Note however, that

TABLE 5. Adjusted dickey-fuller tests for scenario S1.

Simulation	p-value	Stat.	Mean	Std. Dev.
S1:0	5.707×10^{-9}	−6.631	26.73	4.21
S1:2	1.391×10^{-9}	−6.886	26.34	4.17
S1:4	4.596×10^{-9}	−6.671	26.69	4.20
S1:6	1.305×10^{-9}	−6.898	26.32	4.16
$X_{4.8}$	6.058×10^{-9}	−6.620	26.84	4.20
Simulation	p-value	Stat.	Mean	Std. Dev.
S1:1	6.397×10^{-7}	−5.737	13.12	2.23
S1:3	4.614×10^{-7}	−5.802	12.88	2.21
S1:5	6.169×10^{-7}	−5.744	13.12	2.23
S1:7	4.601×10^{-7}	−5.802	12.88	2.21
$X_{2.4}$	1.874×10^{-8}	−6.413	13.45	2.29

Ground truth $X_{4.8}$ for S1:0/2/4/6 and $X_{2.4}$ for S1:1/3/5/7 and S1:3. With $\lambda_{X_{4.8}} = 4.8\,\frac{ped}{min}$ for $X_{4.8}$ and $\lambda_{X_{2.4}} = 2.4\,\frac{ped}{min}$ for $X_{2.4}$ respectively.

for this simple scenario, variations with the same beacon and map intervals but with different aggregation algorithms (YMF and $\text{yDist}_{0.9/60}$) the null hypothesis of the Mann-Whitney U test cannot be dismissed, $p_{S1:0/4} = 2.596 \times 10^{-1}$ and $p_{S1:2/6} = 5.118 \times 10^{-1}$, thus assuming that these distributions are the same. This holds for both pedestrian frequencies $\lambda_{X_{2.4}} = 2.4\,\frac{ped}{min}$ and $\lambda_{X_{4.8}} = 4.8\,\frac{ped}{min}$ respectively.

3) SUMMARY (S1)

The pedestrian density map with beacon sensing can recreate stationary pedestrian flows. Higher packet transmission frequencies reduce the error count. However, the benefit of the yDist algorithm compared to the YMF algorithm cannot be seen in this simple scenario.

C. STATIONARY SCENARIO (S2)

In the stationary scenario, the reaction time of the density map is investigated when a sudden change in the number of pedestrians is observed. For this, pedestrians are placed

TABLE 6. Parameter variation for scenario S1 with N=20.

Simulation	Δt_{Beacon}	Δt_{Map}	λ	Algorithm
S1:0	300 ms	1000 ms	$4.8\frac{ped}{min}$	$yDist_{0.9/60}$
S1:1	300 ms	1000 ms	$2.4\frac{ped}{min}$	$yDist_{0.9/60}$
S1:2	700 ms	4000 ms	$4.8\frac{ped}{min}$	$yDist_{0.9/60}$
S1:3	700 ms	4000 ms	$2.4\frac{ped}{min}$	$yDist_{0.9/60}$
S1:4	300 ms	1000 ms	$4.8\frac{ped}{min}$	YMF
S1:5	300 ms	1000 ms	$2.4\frac{ped}{min}$	YMF
S1:6	700 ms	4000 ms	$4.8\frac{ped}{min}$	YMF
S1:7	700 ms	4000 ms	$2.4\frac{ped}{min}$	YMF

$TTL_{Beacon} = 5\,s$ and $TTL_{Map} = 30\,s$ for all runs.

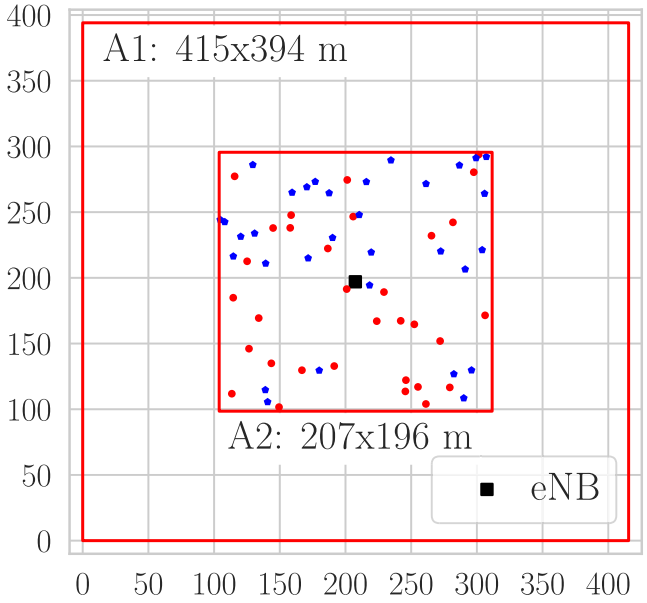


FIGURE 9. Simulation setup for S2 with two areas A1 and A2. At the event time t_e = 50s half of the pedestrian (blue pentagons) will be removed. Figure shows the first seed with 64 pedestrians at the beginning of the simulation.

evenly distributed in x- and y-direction, in an area around a single base station. During the simulation, the pedestrians remain stationary. At a given event time $t_e = 50\,s$, half of the pedestrians are removed. The quantity of interests for this scenario S2 are the convergence time and the relative error in the mean pedestrian count. The convergence time is defined as the time it takes for the pedestrian count of the mean density map $\hat{C}_{ped}$ to reach the correct value, within an error band of ±5%, after the removal event took place.

Two areas are used in which pedestrians are placed: The first area, A1, has a dimension of 415 × 394 m, and the second area, A2, (207 × 196 m) which is 25% of area A1. Both areas are centered on the base station as shown in Fig. 9. To get an understanding of the convergence time, 11 pedestrian densities are simulated with changes in the area (A1 and A2), the pedestrian position (20 positional seeds), as well as changes in the aggregation algorithm (yDist, YMF). The total number of simulations for scenario S2 is $|S2| = 11 \times 2 \times 20 \times 2 = 880$.

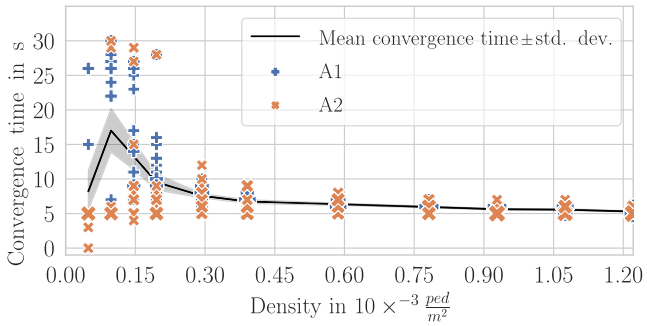


FIGURE 10. Convergence time over density and area size for N=766/880 simulations. The convergence time is the time it takes to reach the new ground truth within an error band of ±5%. 98 simulations never reached the error band before the event time and 16 simulation did not reach the error band within 50 seconds. The marker size represents the number of simulation with same convergence time.

TABLE 7. Number of simulations for scenario S2 not reaching selection criterion.

Density	Area	yDist before	yDist after	ymf before	ymf after
0.000049	A1	16	1	16	1
0.000049	A2	9	0	9	0
0.000098	A1	6	3	6	3
0.000098	A2	9	0	9	0
0.000147	A1	3	1	3	1
0.000147	A2	4	2	4	2
0.000195	A2	2	1	2	1

114/880 simulations conducted for scenario S2 did not reach the selection criterion or did not reach convergence. A simulation is not selected if the map $\hat{C}_{ped}$ for a simulation deviates ±5% at every point in time before the event time in scenario S2 (column *before*). If a simulation was correct at least once before but did not reach the error band of ±5% after 50 seconds, the simulation is deemed to not converge to the new value after the event time.

From the 880 simulations, 98 simulations have not reached a mean pedestrian count before the event time $t_e = 50s$ within an error band of ±5%. These simulations are removed from the analysis because we want to analyze the convergence time of connected density maps. Including these simulations would introduce false positives where simulations of discontented density maps reach fast convergence only because the disconnected pedestrians are removed. Most of the removed simulations have small densities in the larger area A1. This is expected, because the sparsity of agents in a larger area creates disconnected clusters such that each cluster only sees its members, leading to an underestimation of the pedestrian count. For the smaller area A2, this happens less frequently. 16/880 simulations did not reach the error band of ±5% within 50 seconds after the event time t_e. Due to the static scenario, a longer simulation time would not lead to a late convergence and thus it is assumed that these simulations would never converge. The reason for this is again the creation of disconnected clusters due to the removal of agents. Tbl. 7 shows which simulations were removed or did not reach convergence.

Fig. 10 shows the convergence time of the simulation separated by density and area. The used algorithms

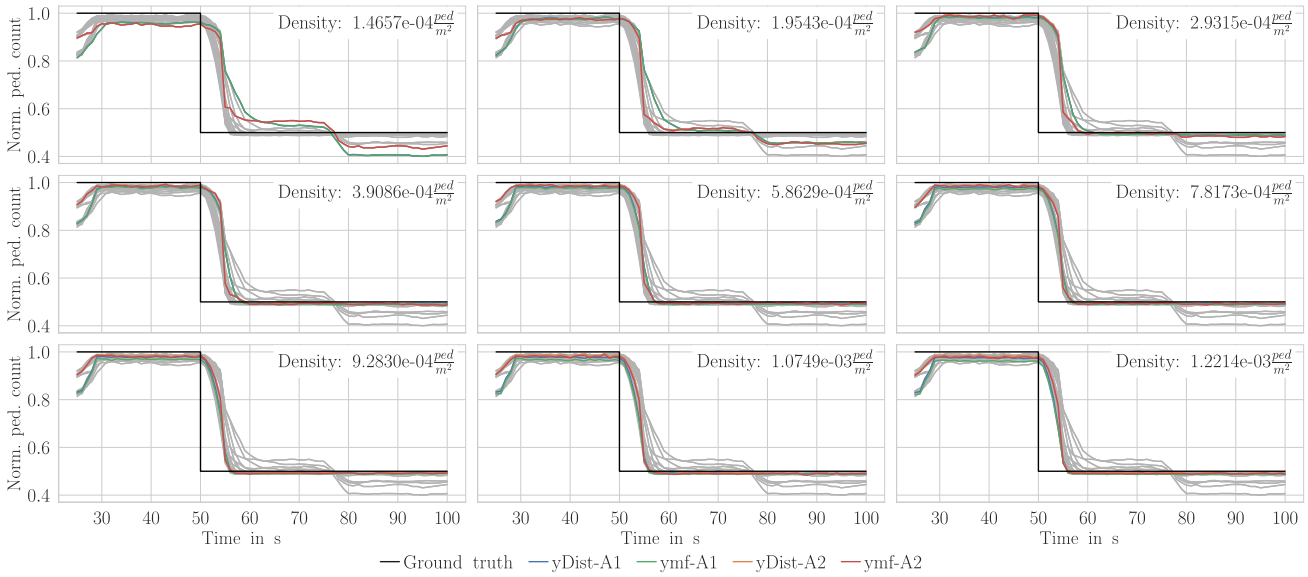


FIGURE 11. Normalized mean map count over time (N=20 receptions with different node positions) for different density values separated by area size (A1, A2) and selection algorithms (yDist, YMF). The figure shows high errors for very lower densities and very high densities. Simulations where agents are placed closer to the eNB (area A2) have continuously smaller count errors. The effect of the aggregation algorithms, in reducing the error, increases with the density. For statistical significance, see Tbl. 8.

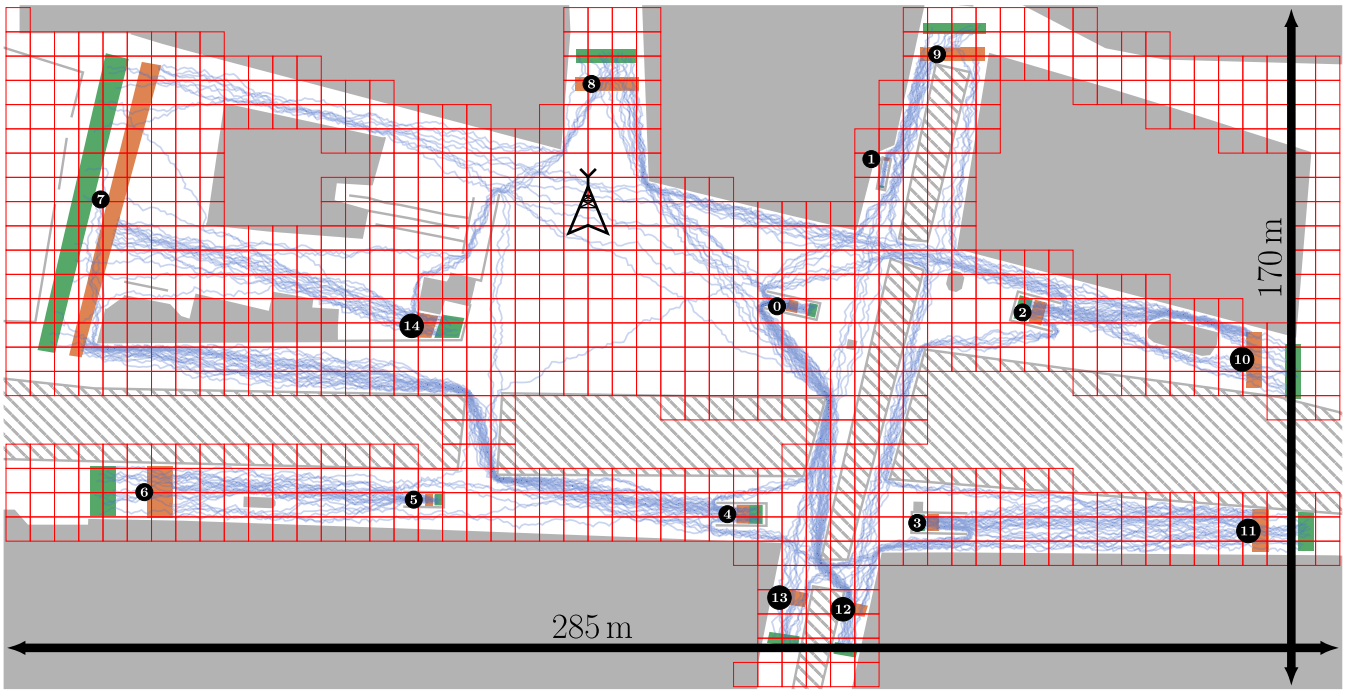


FIGURE 12. Above ground area of subway station "Müncher Freiheit" with cells (red grid). The generation interval is defined by a negative exponential distribution with the parameter $\frac{1}{\lambda} = 25$ s. Agents are routed based on the OD-Matrix shown in Tbl. 9. The trajectories (blue paths) depicted in the figure are the dynamic behavior for one mobility seed. Gray polygons are obstacles (buildings), gray hatched polygons represent streets such that pedestrian will use dedicated crossings.

(yDist, YMF) did not affect the convergence time. The different areas are represented by the marker shape. The size of the marker represents multiple simulations that have the same convergence time. It clearly shows that small density values have a larger convergence time with a significantly larger variation. A possible reason for the longer convergence times is that the communication graph between the pedestrians is sparse such that density information has to be communicated over multiple hops. This also explains that larger areas with the same density have considerably larger convergence times. After a limit density of around 0.0003 $\frac{ped}{m^2}$, the mean convergence time for all simulations is 5.92 s with a standard deviation of 0.87 s.

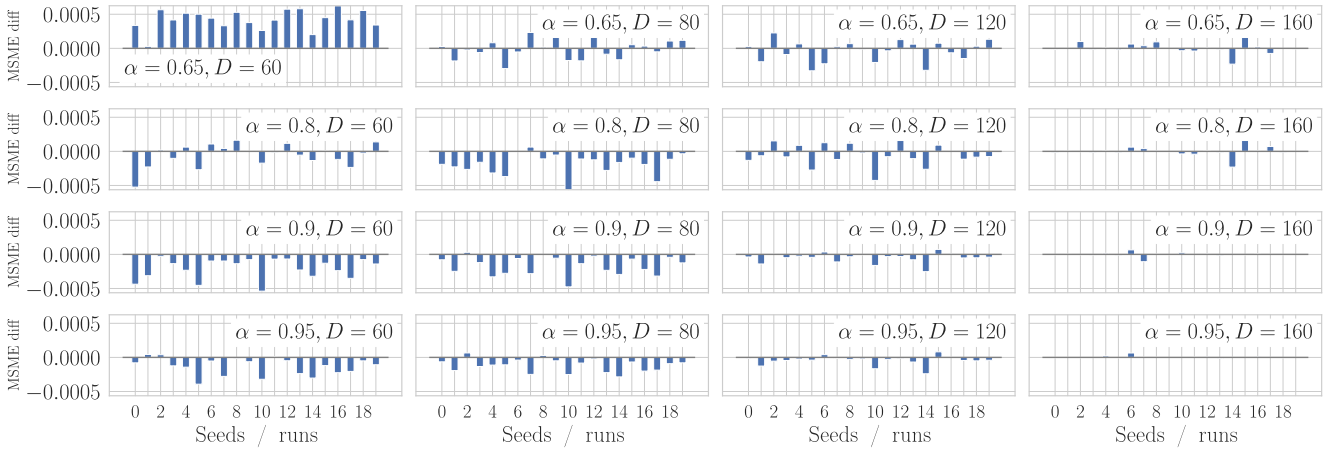


FIGURE 13. Parameter study with scenario S3 showing the MSME difference between the yDist and YMF algorithm for all N=20 seeds.

The second quantity of interest is the relative error in the mean pedestrian count as shown in Fig. 11. It shows the $\hat{C}_{ped}(t)$ for each simulation separated by density, area, and algorithm and averaged over N=20 positional seeds for each (density, area, algorithm) triplet. The normalized mean count error is shown by the y-distance between the $\hat{C}_{ped}(t)$ time series and the ground truth (black line). The error decreases with rising density values to a point and then increases again for higher densities. Also with rising density values the difference in the applied algorithms increase such that the yDist algorithm performs better in denser situations. This improvement is clearer for the larger area (A1) and densities above $9.283 \times 10^{-4} \frac{ped}{m^2}$. The gray lines in each grid plot show the normalized mean pedestrian count of the other densities to ease comparison between densities.

1) SUMMARY (S2)

The scenario shows that the reaction time of the pedestrian density map to changes in the number of agents present depends foremost on the time to live τ_{NT} of the neighborhood table which encodes the beacon sensing of pedestrian density maps. The effect of pedestrian density is less prevalent compared to the distance between nodes. This can be seen in simulations with the same density where considerably different convergence times are observed when the size of the area changes. Furthermore, the yDist algorithm shows better performance in the stationary example with increased density.

D. DYNAMIC SCENARIO S3

We use a real scenario concerning the area around the subway station "Müncher Freiheit", Munich, Germany, to perform a parameter study to find the parameters α and D needed for the yDist algorithm. The real-world scenario is modeled in the Vadere simulator to generate the mobility traces needed for the simulations. Fig. 12 shows the Vadere topography with an overlay representing density map cells used for the scenario. We only consider cells that can be reached by agents during the simulation. Cells completely covered by obstacles, such as buildings or roads, are removed from the analysis. Note that all removed cells did not contain any errors, meaning the measurement value of the removed cells is always zero and these cells never had any erroneous counts during the simulation time.

TABLE 8. Comparison of norm. mean map count for S2.

Density	A1 p-value		A2 p-value	
0.000147	0.7429	H0 (same)	1.0000	H0 (same)
0.000195	0.6385	H0 (same)	0.8682	H0 (same)
0.000293	0.2533	H0 (same)	0.6797	H0 (same)
0.000391	0.0363	H1 (diff)	0.3037	H0 (same)
0.000586	0.0015	H1 (diff)	0.2136	H0 (same)
0.000782	0.0007	H1 (diff)	0.1303	H0 (same)
0.000928	0.0002	H1 (diff)	0.0138	H1 (diff)
0.001075	2.2768×10^{-5}	H1 (diff)	0.0083	H1 (diff)
0.001221	5.3884×10^{-6}	H1 (diff)	0.0018	H1 (diff)

Mann-Whitney U test with a confidence of 5%. The null hypothesis states that the distributions of the two samples, mean normalized pedestrian count for the different algorithms, are from the same distribution. Otherwise, the distributions differ.

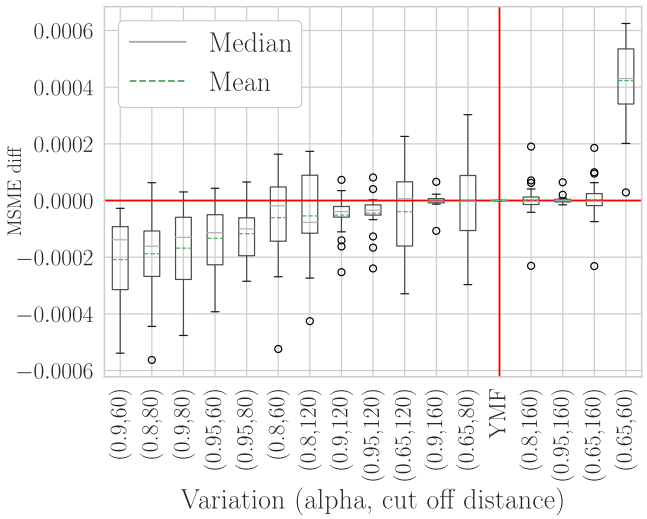
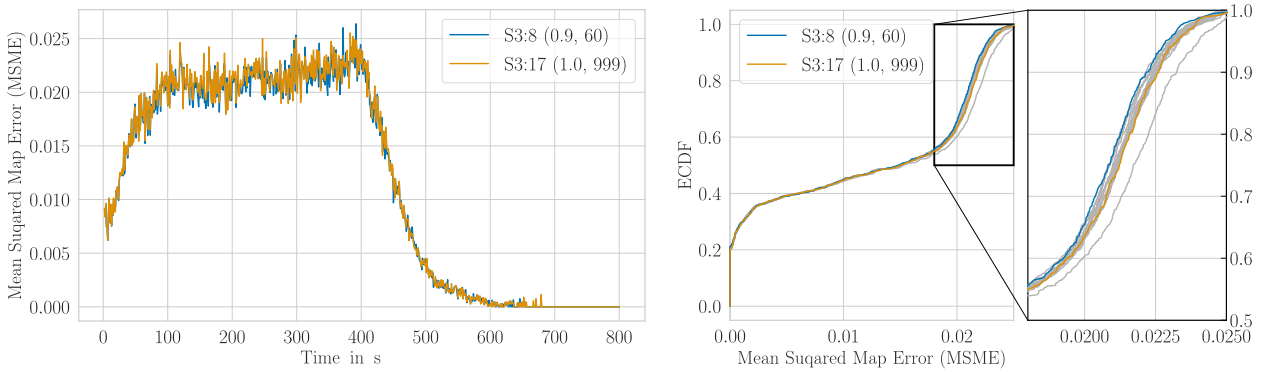


FIGURE 14. MSME difference sorted by mean.



(a) MSME time series.

(b) Empirical cumulative density function of MSME of all variations. S3:8 and S3:17 are highlighted.

FIGURE 15. **Comparison of best variation of yDist with YMF algorithm. The S3:8 variation has better performance in cases with higher MSME compared to the YMF (S3:17) variation.**

TABLE 9. **OD-Matrix used for Scenario S3.**

	0	1	2	3	4	5	6	7	8	9	10	11	12	14
0	·	·	·	·	·	·	·	·	·	·	1.00	·	·	·
1	·	·	·	·	·	·	·	·	·	1.00	·	·	·	·
2	·	·	·	·	·	·	·	·	·	·	1.00	·	·	·
3	·	·	·	·	·	·	·	·	·	·	·	1.00	·	·
4	·	·	·	·	·	·	·	1.00	·	·	·	·	·	·
5	·	·	·	·	·	·	1.00	·	·	·	·	·	·	·
6	·	·	·	·	0.50	0.50	·	·	·	·	·	·	·	·
7	0.20	·	0.20	0.20	0.20	·	·	·	·	·	·	·	·	0.20
8	0.25	·	·	·	0.25	·	·	0.25	·	·	·	·	·	0.25
9	0.17	0.17	0.17	·	0.17	0.17	·	·	·	·	·	·	·	0.17
10	0.17	0.17	0.17	·	·	·	·	0.17	0.17	0.17	·	·	·	·
11	0.25	·	·	0.25	·	·	·	·	0.25	·	·	·	0.25	·
12	0.20	·	0.20	0.20	0.20	·	·	·	·	0.20	·	·	·	·
13	0.20	·	0.20	0.20	0.20	·	·	·	·	0.20	·	·	·	·
14	·	·	·	·	·	·	·	1.00	·	·	·	·	·	·

See Fig. 12 for origin/destination markers (circled numbers).

For the simulation, pedestrians are spawned in sources (green polygons). At creation time, each agent is assigned to one target (orange polygons) based on the distribution given by the origin-destination (OD) matrix shown in Tbl. 9. Each element contains the probability that an agent spawned at the source will go to the respective target. For example, an agent created at source ❼ will go to target ❹ with a probability of 20%. The origin/destination numbers are shown in Fig. 12. The OD-matrix is constant throughout the simulation study. However, due to the distribution-based OD matrix, the overall pedestrian dynamic will change considerably with different seeds.

For the parameter study, the parameter space for α and the cutoff distance D is sampled using a full grid approach. The parameter α is sampled at [0.65, 0.80, 0.90, 0.95] and the cut of distance D at [60, 80, 120, 160] meters, which will lead to 16 parameter variations. Parameter variations for the yDist algorithm with α values below 50% show consistently worse results in initial studies and are therefore not listed. To allow a comparison to the youngest measurement first algorithm, an additional variation is included where the parameter α is set to 1.0, which will reduce the yDist algorithm to the YMF algorithm. Compare rank function definitions (11) and (18). Each parameter variation is simulated with N=20 seed such that for scenario $|S3| = 17 \times 20 = 340$ simulations are conducted. For this scenario, we reduced the time to live $\tau_{Map} = 15.0$ s of map measurements as this reduces the pedestrian count error in this dynamic situation. This only reduced the absolute value of MSME in the parameter study.

For scenario S3, we compare the mean squared map error (MSME), to find the parameter variation of the yDist algorithm which provides better or similar performance compared to the YMF algorithm. Fig. 13 depicts the MSME difference for all variations S3:0-15 (yDist) compared to the youngest measurement first. The right column of Fig. 13 shows that with a distance parameter of $D = 160$ m there is no or very little difference between the yDist and YMF algorithms. With decreasing distance and increasing α values, the yDist algorithm shows consistently lower mean squared map errors in all runs. The variation S3:8 with ($\alpha = 0.9, D = 60$) has the biggest mean difference, as shown in Fig. 14. The box plot marks variation S3:17 YMF, with which all other variations are compared. Comparing the variation S3:8 with YMF over time, Fig. 15a shows a slightly reduced error in the interval of 100 to 400 seconds. In the start and end phases of the simulation, the errors look similar. Fig. 15b shows the empirical CDF for all variations, with S3:8 (yDist) and S3:17 (YMF) highlighted. For small MSME values, all variations behave similarly. With increasing MSME values, for yDist, all but one variation do outperform the YMF. However, the performance gain is smaller compared to the gains seen in the stationary Scenario S2.

1) SUMMARY (S3)

The pedestrian density map shows good performance for a real-world scenario. Both proposed heuristics show similar performance with a maximum MSME of 0.025, which translates to an average pedestrian count error for an average cell of $\sqrt{0.025} = \pm 0.158\,113\,883\,008\,419\,0$.

VII. CONCLUSION AND FUTURE WORK

We presented a generalized architecture for a decentralized pedestrian measurement map (DPMM) to sense, disseminate, and aggregate arbitrary measurement quantities without the need for centralized services. For this, we defined a set-based

structure containing pairs of the form measurement quantity and associated metadata sets. Based on this we, defined abstract rules to generate aggregated measurement maps in a node-local manner. Specific algorihms for aggregation and strategies to create a cell transmission order are embedded in the abstract architecture with the possibility to exchange the implementation as needed.

Furthermore, we demonstrated that the mobility patterns of the sensing agents (i.e. pedestrians) play a major role in the achievable measurement accuracy. An important metric to determine this accuracy is the distribution of the empty interval length for cells because only when an agent is within sensing range, measurements are possible. The proposed metrics to capture the cell's empty interval length can be recorded locally by each node, thus allowing it to be used as a parameter to configure dynamic variables such as dissemination interval length or cell transmission order.

For the use case of pedestrian density measurements, we introduced a beacon-based sensing model that utilizes active position beacons and the neighborhood table structure to count pedestrians in the direct vicinity. We demonstrated that the beacon sensing approach is capable to recreate stationary flows of pedestrians. In stationary scenarios, the time-to-live τ_{NT} of neighborhood table entries has a significant influence on the convergence time, i.e. the time required for getting accurate measurements when the number of pedestrians has changed drastically. Furthermore, the inter-node distance has a stronger effect on the dissemination of new information than the pedestrian density itself. For the aggregation, we selected two simple heuristics to aggregate received foreign maps locally. Detailed event-based simulations of several scenarios indicate that with these heuristics an accurate estimate of the pedestrian density within the local area can be obtained. The heuristics both produce similar results regarding observed differences based on node mobility. In future work, we plan to analyze how additional contextual information, besides age and sensing distance, can help to increase measurement accuracy.

A known limitation of the proposed decentralized pedestrian measurement maps is the fixed inter-transmission interval, which can lead to congestion of the radio channel in high-density situations or high background traffic. To mitigate this, congestion control mechanisms can be introduced as well as mobility-based algorithms that take into account the perceived mobility to dynamically adapt beacon and map transmission intervals. Furthermore, the selection process of cells for transmission can be extended to consider mobility or measurement quantity dynamics. These potential improvements of the presented approach will also be evaluated in detail in future work.

REFERENCES

[1] C. M. Mayr and G. Köster, "Guiding crowds when facing limited compliance: Simulating strategies," *PLoS ONE*, vol. 17, no. 11, pp. 1–24, Nov. 2022, doi: 10.1371/journal.pone.0276229.

[2] S. Schuhbäck and L. Wischhof, "Decentralized pedestrian density maps based on sidelink communication," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6, doi: 10.1109/ICCWorkshops50388.2021.9473545.

[3] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–31, Sep. 2015, doi: 10.1145/2794400.

[4] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, Apr. 2019, doi: 10.1109/COMST.2019.2914030.

[5] J. Phuttharak and S. W. Loke, "A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered Internet-of-Things," *IEEE Access*, vol. 7, pp. 304–324, 2019, doi: 10.1109/ACCESS.2018.2885353.

[6] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011, doi: 10.1109/MCOM.2011.6069707.

[7] A. I. Middya and S. Roy, "Spatial interpolation techniques on participatory sensing data," *ACM Trans. Spatial Algorithms Syst.*, vol. 7, no. 3, pp. 1–32, Sep. 2021, doi: 10.1145/3457609.

[8] P. Zhou, Y. Zheng, and M. Li, "How long to wait? Predicting bus arrival time with mobile phone based participatory sensing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1228–1241, Jun. 2014, doi: 10.1109/TMC.2013.136.

[9] E. K. Tokuda, Y. Lockerman, G. B. A. Ferreira, E. Sorrelgreen, D. Boyle, R. M. Cesar-Jr, and C. T. Silva, "A new approach for pedestrian density estimation using moving sensors and computer vision," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 4, pp. 1–20, Dec. 2020, doi: 10.1145/3397575.

[10] J. De Benedetto, P. Bellavista, and L. Foschini, "Proximity discovery and data dissemination for mobile crowd sensing using LTE direct," *Comput. Netw.*, vol. 129, pp. 510–521, Dec. 2017, doi: 10.1016/j.comnet.2017.08.002.

[11] C. Costa, C. Laoudias, D. Zeinalipour-Yazti, and D. Gunopulos, "SmartTrace: Finding similar trajectories in smartphone networks without disclosing the traces," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 1288–1291, doi: 10.1109/ICDE.2011.5767934.

[12] M. Marjanovic, A. Antonic, and I. P. Žarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10662–10674, 2018, doi: 10.1109/access.2018.2799707.

[13] L. Lin, X. Lin, and X. Wang, "A mobile crowd sensing ecosystem based on fog computing infrastructure," in *Proc. 20th Int. Conf. Ubiquitous Comput. Commun. (IUCC/CIT/DSCI/SmartCNS)*, Dec. 2021, pp. 108–115, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00030.

[14] A. Bazzi, A. O. Berthet, C. Campolo, B. M. Masini, A. Molinaro, and A. Zanella, "On the design of sidelink for cellular V2X: A literature review and outlook for future," *IEEE Access*, vol. 9, pp. 97953–97980, 2021, doi: 10.1109/ACCESS.2021.3094161.

[15] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. Standard EN 302 637–2 V1.3.1, ETSI, Sep. 2014.

[16] *Intelligent Transport System (ITS);Vulnerable Road Users (VRU) awareness;Part 2: Functional Architecture and Requirements definition; Release 2*. Standard TS 103 300–2 V2.2.1, ETSI, 2021.

[17] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*, Standard EN 302 895 V1.1.1, ETSI, Sep. 2014.

[18] *Itelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2*. Standard TR 103 562 V2.1.1, ETSI, Dec. 2019.

[19] F. A. Schiegg, I. Llatser, D. Bischoff, and G. Volk, "Collective perception: A safety perspective," *Sensors*, vol. 21, no. 1, p. 159, Dec. 2020, doi: 10.3390/s21010159.

[20] Q. Delooz, A. Willecke, K. Garlichs, A.-C. Hagau, L. Wolf, A. Vinel, and A. Festag, "Analysis and evaluation of information redundancy mitigation for V2X collective perception," *IEEE Access*, vol. 10, pp. 47076–47093, 2022, doi: 10.1109/ACCESS.2022.3170029.

[21] A. Willecke, K. Garlichs, F. Schulze, and L. C. Wolf, "Vulnerable road users are important as well: Persons in the collective perception service," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2021, pp. 24–31, doi: 10.1109/VNC52810.2021.9644669.

[22] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017, doi: 10.1109/COMST.2017.2745201.

[23] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu, "CrowdNet," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 640–644, doi: 10.1145/2964284.2967300.

[24] B. Yogameena and C. Nagananthini, "Computer vision based crowd disaster avoidance system: A survey," *Int. J. Disaster Risk Reduction*, vol. 22, pp. 95–129, Jun. 2017, doi: 10.1016/j.ijdrr.2017.02.021.

[25] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," *Pattern Recognit. Lett.*, vol. 107, pp. 3–16, May 2018, doi: 10.1016/j.patrec.2017.07.007.

[26] J. Huo, X. Fu, Z. Liu, and Q. Zhang, "Short-term estimation and prediction of pedestrian density in urban hot spots based on mobile phone data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10827–10838, Aug. 2022, doi: 10.1109/TITS.2021.3096274.

[27] M. Wirz, T. Franke, D. Roggen, E. Mitleton-Kelly, P. Lukowicz, and G. Troster, "Inferring crowd conditions from Pedestrians' location traces for real-time crowd monitoring during city-scale mass gatherings," in *Proc. IEEE 21st Int. Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises*, Jun. 2012, pp. 367–372, doi: 10.1109/WETICE.2012.26.

[28] P. Torkamandi, L. Pajevic Kärkkäinen, and J. Ott, "Characterizing Wi-Fi probing behavior for privacy-preserving crowdsensing," in *Proc. Int. Conf. Model. Anal. Simul. Wireless Mobile Syst. Int. Conf. Model. Anal. Simul. Wireless Mobile Syst.*, Oct. 2022, pp. 203–212, doi: 10.1145/3551659.3559039.

[29] A. Vial, W. Daamen, A. Y. Ding, B. van Arem, and S. Hoogendoorn, "AMSense: How mobile sensing platforms capture Pedestrian/Cyclist spatiotemporal properties in cities," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 1, pp. 29–43, Jan. 2022, doi: 10.1109/MITS.2019.2953509.

[30] (2022). INET. *INET Framework—Open-Source OMNeT++ Model Suite for Wired, Wireless and Mobile Networks*. [Online]. Available: https://inet.omnetpp.org/

[31] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G—An OMNeT++ library for end-to-end performance evaluation of 5G networks," *IEEE Access*, vol. 8, pp. 181176–181191, 2020, doi: 10.1109/ACCESS.2020.3028550.

[32] R. Riebl, H.-J. Gunther, C. Facchi, and L. Wolf, "Artery: Extending veins for VANET applications," in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Jun. 2015, pp. 450–456, doi: 10.1109/MTITS.2015.7223293.

[33] C. Sommer, D. Eckhoff, A. Brummer, S. D. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins: The open source vehicular network simulation framework," in *Recent Advances in Network Simulation*. Cham, Switzerland: Springer, 2019, pp. 215–252, doi: 10.1007/978-3-030-12842-5.

[34] P. A. Lopez, E. Wiessner, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, and P. Wagner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582, doi: 10.1109/ITSC.2018.8569938.

[35] L. Wischhof, M. Kilian, S. Schuhbäck, and G. Köster, "On the influence of microscopic mobility in modelling pedestrian communication," in *Proc. 7th Int. Symp. Ubiquitous Netw. (UNet)*, 2022, pp. 1–9.

[36] B. Kleinmeier, B. Zönnchen, M. Gödel, and G. Köster, "Vadere: An open-source simulation framework to promote interdisciplinary understanding," *Collective Dyn.*, vol. 4, pp. 1–18, Sep. 2019, doi: 10.17815/CD.2019.21.

[37] S. Schuhbäck, N. Daßler, L. Wischhof, and G. Köster, "Towards a bidirectional coupling of pedestrian dynamics and mobile communication simulation," in *Proc. OMNeT++ Community Summit*, vol. 66, 2019, pp. 60–67, doi: 10.29007/nnfj.

[38] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An interface for coupling road traffic and network simulators," in *Proc. 11th Commun. Netw. Simul. Symp.*, Apr. 2008, pp. 155–163, doi: 10.1145/1400713.1400740.

[39] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. 3rd Int. Conf. Simul. Tools Techn. (ICST)*, 2010, pp. 1–10, doi: 10.4108/ICST.SIMUTOOLS2010.8684.

[40] *Propagation Data and Prediction Methods for the Planning of Shortrange Outdoor Radiocommunication Systems and Radiolocal Area Networks in the Frequencyrange 300 MHz to 100 GHz*, Standard ITU-R P.1411–6, Int. Telecommunication Union, 2012.

[41] G. Nardini, G. Stea, A. Virdis, and D. Sabella, "Simu5G: A system-level simulator for 5G networks," in *Proc. 10th Int. Conf. Simul. Model. Methodologies, Technol. Appl.*, 2020, pp. 68–80, doi: 10.5220/0009826400680080.

STEFAN SCHUHBÄCK (Graduate Student Member, IEEE) received the B.Eng. degree from the Munich University of Applied Science (HM), the M.Sc. degree in industrial engineering and management from Friedrich Alexander University, Erlangen–Nürnberg, and the M.Sc. degree in computer science from HM. Currently, he is pursuing the Dr.rer.nat. degree in computer science with a Cooperative Doctorate Program with TUM and HM.

LARS WISCHHOF (Senior Member, IEEE) received the Dipl.-Ing. degree in computer engineering from Technische Universität Berlin and the Dr.-Ing. degree from the Institute of Communications, Technische Universität Hamburg, Harburg, Germany. He is a Professor with the Department of Computer Science and Mathematics, Munich University of Applied Sciences, Munich. His research interest includes applied mobile communication, in particular vehicular and pedestrian communication. He is a member of VDE and a Senior Member of ACM.

JÖRG OTT (Member, IEEE) received the Diploma degree in computer science from TU Berlin, in 1991, the Diploma degree in industrial engineering from TFH Berlin, in 1995, and the Ph.D. degree in computer science from TU Berlin, in 1997. He has been the Chair of connected mobility with the Faculty of Informatics, Technische Universität München, since August 2015. He explores edge and in-network computing as well as decentralized services. His research interests include network architectures, (transport) protocols, measuring and modeling human mobility, internet usage as a basis for design and evaluation, and algorithms for connecting mobile nodes to the internet and to each other, covering the spectrum from delay-tolerant to real-time networking.