DOI: 10.15439/2020F189
ISSN 2300-5963 ACSIS, Vol. 21

Integrated Human Tracking Based on Video and Smartphone Signal Processing within the Arahub System

Jan Ludziejewski, Łukasz Grad
Uniwersytet Warszawski
Email: {jan.ludziejewski, lukasz.grad}@mimuw.edu.pl

Łukasz Przebinda
Arahub & Myled
Email: l.przebinda@myled.pl

Tomasz Tajmajer
QED Software
Email: tomasz.tajmajer@qed.pl

***Abstract*—Embedded platforms with GPU acceleration, designed for performing machine learning on the edge, enabled the creation of inexpensive and pervasive computer vision systems. Smartphones are nowadays widely used for profiling and tracking in marketing, based on WiFi data or beacon-based positioning systems. We present the Arahub system, which aims at integrating world of computer vision systems with smartphone tracking for delivering data useful in interactive applications, such as interactive advertisements. In this paper we present the architecture of the Arahub system and provide insight about its particular elements. Our preliminary results, obtained from real-life test environments and scenarios, show that the Arahub system is able to accurately assign smartphones to their bearers, based on visual and WiFi/Bluetooth positioning data. We show the commercial value of such system and its potential applications.**

I. Introduction

WHILE video monitoring systems are currently found everywhere, still, most of them are used for security applications. Systems installed in commercial zones, stores or cafes, could deliver valuable information to owners of such places, yet automatic analysis of such data requires advanced computer vision systems. Embedded platforms with GPUs for providing machine learning to the edge, enabled the creation of inexpensive and pervasive devices, that may process high-level data extracted from video streams.

As virtually every person is equipped with a smartphone these days, many companies are offering analytic services based on location tracking and mobile applications. Location-based marketing, geofencing or predictive analysis are all more widely used for companies to deliver personalized, targeted marketing. Yet this source of data has its limitations - it is difficult to deliver real-time information about a person which is at a particular place - and this is crucial if one wants to provide personalization and interaction, e.g. a dedicated advertisement displayed to a specific person.

In this work we present the Arahub project. It is focused on combining the world of computer vision systems with smartphone tracking for delivering data useful in interactive applications, where both location and profile of a person are required. The primary use-case for Arahub is digital marketing system that could be used for marketing campaigns delivered to specific persons at specific places.

In this paper the overall architecture of the Arahub system is described. We provide insights into particular elements of the system and methods used. We also present preliminary results, which we were able to obtain in real-life environments.

A. Principle of operation

The primary goal of Arahub is to provide statistical data about people present near an area of interest. Examples of such data are: the number of people watching a commercial on a display withing a specified time period, the gender of a person currently watching a shop exposition, shopping preferences of a person moving towards a display, etc. Such statistics may be based on data gathered from several sources: vision systems [1], [2], [3], indoor-positioning [4], [5] or mobile apps. The most interesting (and challenging) is the possibility of integrating data from multiple sources [6] to gather even more commercially valuable insight.

Let us consider the following scenario: a person with a smartphone has a loyalty application installed and running. This person is shopping in a store that is supported by the loyalty application. The owner of the store may have access to data provided by the application, such as the purchase history of the given customer. The owner, however, cannot directly match that data with a particular person currently visiting the store, as localization data may be too coarse. Yet, the owner of a store has access to a visual monitoring system, which could be used for precise visual tracking of all customers. Those two data sources, when properly linked together, could provide rich data attributed to a particular person currently visiting the store. Such a link could be established by combining the position of a person based on visual cues with the position of the mobile device owned by that person.

B. Motivation

Digital Out Of Home (DOOH) is a segment of marketing that is based on digital forms of advertising placed outdoors or in indoor public locations (out-of-home). The set of media types, including displays, LED screens and similar, used in DOOH, are referred to as Digital Signage (DS).

As DOOH and DS systems are becoming more common, there is a need for novel methods of targeting, interaction and content design, that could use the potential of this new type of

Supported by Innoventure and NCBiR (POIR.01.03.01-00-0022/16)

advertising. A particularly interesting ideas may be borrowed from the world of online advertising, which after decades of existence has become a mainstream advertising channel.

Existing DOOH systems are passive in terms of targeting - marketing content is selected based on long-term demography statistics or, in the best case, on custom surveys made for a particular location. It is obvious that such methods of audience analysis could not be compared to precise on-line targeting based on browser cookies or shopping history. Yet there is a high potential for using external data sources in DOOH. Such cases, using traffic or weather data, are already existing.

The biggest potential is in so-called "programmatic DOOH", which envisions a novel method of selling DOOH media - not by air time or by surface area, but by the number of views, or even views of the specified audience with particular interests or shopping history. To enable such operation, one needs to provide real-time data about the audience or particular viewers. Arahub is meant to provide such functionality and connect the advertising from online world with digital media existing in the real world.

Even though real-time, personalized DOOH is the main motivation behind the development of Arahub, there are many other, useful applications of such a system. The integration of multi-modal data sources for more accurate positioning and profiling may be used in smart-city and smart-home [7] environments, especially in healthcare or public services [8]. Also, security systems could benefit from more accurate analysis methods; facial recognition methods - despite rising privacy concerns - may also provide valuable insight if used with respect to legal regulations [9]. Finally, a system such as Arahub is a source of meta-data that could be used to learn about general behaviors and trends in the society, which can be used for making predictive models or inferring rules [10].

II. Related work

Positioning Systems based on WiFi and Bluetooth signals have been an active area of research over the last years. The two most common approaches to device localization based on a system of multiple WiFi access points or Bluetooth beacons are triangulation and fingerprinting.

Triangulation methods can be further divided into lateration and angulation [11]. These methods use the estimated distance from several transmitters or receivers based on signal attenuation [12], time characteristics of the propagated signal, e.g. Time of Arrival [13], Time Difference of Arrival [14] or are based on the direction of the received signal - Angle of Arrival [15]. Triangulation methods achieve good results in open space environments. However, they perform significantly worse in the indoor conditions where the signals may be reflected by several obstacles and there is no clear line-of-sight between the transmitting and receiving devices.

Fingerprinting methods work in two phases. In the first learning phase, a database of the signal characteristics at known locations is built [16], usually based on the Received Signal Strength Indicator (RSSI). This reference data set is then used in the second stage to perform localization, by comparing the measured signal characteristics with the fingerprints stored in the database. Several methods that improve on the standard fingerprint-based methods have been developed, e.g. statistical post-processing methods to estimate a continuous distribution of RSSI values based on Gaussian Process Theory [17] [18] or parametric estimation of the RSSI distribution [19]. Moreover, [20] presents a comparison between WiFi and Bluetooth localization system based on the fingerprinting approach and shows the advantages of BLE-based localization

In our work, we present a uniform approach for WiFi and Bluetooth signal modeling and develop two methods for estimating RSSI distribution along with a probabilistic Indoor Positioning System. The first approach is based on an extension of the Log-distance path loss model [21], the second method is based on a probabilistic fingerprinting-based model.

The two most common approaches for human tracking using video stream data are neural network based with subsequent box matching and motion detection. Motion detection can be further divided into Background Subtraction, Frame Differencing, Optical Flow and Temporal Differencing [22]. We utilize both approaches, in the second case merging Background Subtraction and Frame Differencing with a custom clustering method. However, multi-camera human tracking generally focuses on Probabilistic Occupancy Maps [23], developing a number of color-based or location-based techniques [24], while we propose a graph-based approach focused on location path similarity without dividing location space into clusters.

III. Arahub system overview

The architecture of the Arahub system consists of: a) distributed sensor network, which includes all equipment installed on-site; b) centralized data aggregation part, which includes multiple services running in the cloud environment. The overview of the architecture is presented in figure 1.

The distributed part of Arahub is based on a custom hardware solution - the Arabox, which integrates a vision system, WiFi monitoring hardware and GPU-enabled computing. In a typical scenario, several Araboxes are installed in one location for precise monitoring of a given point of interest. Moreover, Bluetooth Low Energy (BLE) beacons are also used to enhance the precision of indoor positioning. Araboxes provide high-level data about persons visible by the camera, such as their position on a 2D plane, they also provide the RSSI for WiFi clients connected to a specified WiFi Access Point (WiFi AP).

The data aggregation part has several functions. First of all, it provides interfaces for collecting the data from Araboxes and mobile applications, secondly it runs dedicated algorithms for filtering and combining multi-modal data, and finally, it provides services for accessing and interacting with the data. Arahub system also includes web services for management, visualization and diagnostics.

Another important elements of Arahub are the mobile devices carried by people in monitored locations. Arahub provides two methods for smartphone positioning: a) active - when the smartphone has a dedicated application running,

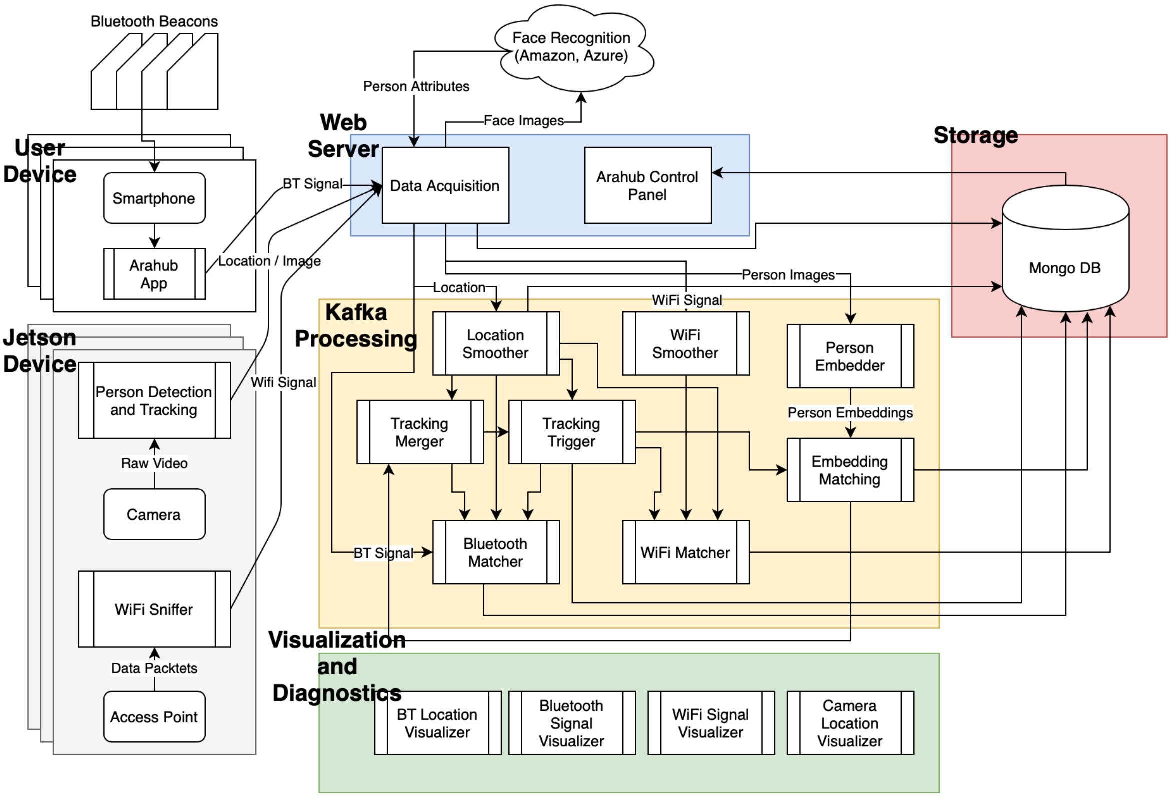

Fig. 1. Arahub architecture overview. A distributed sensor network is based on the Arabox devices installed on-site as well as mobile devices running dedicated Arahub application. Data from the sensor network is sent to the webserver and processed using a data acquisition module. We use Amazon and Microsoft Azure face recognition systems to enrich video data with personal attributes such as age and gender. Then, raw signal and location data are processed within a Data Aggregation system based on Kafka processing engine. We utilize Kafka connectors to save data in a Mongo database for the purpose of business analysis and model training. Arahub system also provides a number of visualization and diagnostics tools that enable monitoring of raw radio signals and locations received as well as tracking and signal-based indoor positioning systems.

b) passive - when the smartphone is connected to a dedicated WiFi network. The details of the operation of those methods are covered in section IV-E

A. Arabox - embedded platform for video and WiFi analysis

Arabox is a dedicated platform for gathering video streams and WiFi analysis. The goal of Arabox design was to create a compact, standalone device, that could locally perform computer vision tasks such as object detection. The device is meant to be installed in commercial zones, with no requirements as to existing infrastructure other than internet connectivity.

At the design stage, two main use-cases of Arabox were taken into consideration: 1) to be installed next to digital displays, where it could provide contextual information about the audience, 2) to be installed in passages such as corridors or stairways in commercial zones, where it would provide information about people visiting certain points of interests. For this reason, two versions of Arabox were developed: a large version (presented in figure 2), with two wide angle cameras integrated into a single enclosure, and a smaller version, with a single camera detached from the main enclosure.

In terms of the hardware platform, both versions of Arabox consist of the same elements. The core is an nvidia's Jetson Nano platform, with 4GBs or RAM and an integrated GPU, capable of CUDA operations. The video stream is provided by an RGB camera with dedicated optics, capable of recording full HD video at 30fps with low noise and in low light conditions. The third part is the WiFi adapter with an antenna dedicated for WiFi monitoring in 2,4GHz and 5GHz bands. Each Arabox also has a proper power adapter and ventilation system included. The enclosure of Arabox in the large version fits all elements inside and is waterproof, thus is suitable for outdoor installation. In this version, two cameras are placed such that their combined field of view angle is not less than 120 degrees. The cameras can be configured for different view angles if needed. The small version is dedicated for indoor installation - a single camera and WiFi adapter with an antenna are enclosed together separately from the Jetson Nano board. Both versions of Arabox have a dedicated mounting system, that allows for mounting to a ceiling or a wall.

The Arabox's embedded system - the Jetson Nano - is running a Linux system with custom software. The software

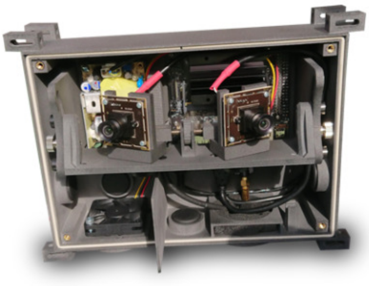

Fig. 2. Arabox prototype - the large version. A custom casing includes all elements: two cameras, Jetson Nano board, WiFi adapter, power supply and cables.

consists of three parts: video processing, WiFi processing and management.

Video processing is done in several steps: first, the raw data from the camera is normalized and throttled, to obtain a stable stream of video images. The stream may be then processed by several algorithms for object detection, such as GPU-based convolutional neural networks (described in more detail in section IV-B). The outputs of those algorithms are bounding boxes, based on which physical 2D positions of objects are calculated. Finally, the calculated positions are sent to the data aggregation system. Depending on the configuration, cropped images of detected objects may be also sent to the data aggregation system for further analysis, e.g. gender detection.

WiFi processing is based on monitor capabilities of an IEEE 802.11ac interface. The WiFi interface is configured to monitor data on channels used by a dedicated Access Point. The software reads control packets sent between that AP and all connected clients in range. It provides the RSSI (Received Signal Strength Indication) of the signal sent by clients, measured in the point where particular Arabox is installed. This data, containing the client's identifier, timestamp and RSSI is then forwarded to the data aggregation system.

A management system is used to provide software updates, configuration changes and to monitor the state of an Arabox. It is based on third-party software, that provides a centralized system for remote management of multiple devices with various internet connectivity (e.g. using third-party, NAT or cellular connections).

Arabox works in a semi-autonomic way - most data processing is done locally, so only high-level data is sent to the data aggregation system. Arabox needs to have constant internet connectivity, however as the data footprint is low, even cellular connections could be used for that purpose.

B. Mobile application

Arahub system uses a custom application developed for Android and iOS systems. The primary goal of this application is to enable indoor positioning based on BLE beacons. The application operates as follows: first, the application listens for familiar beacons IDs in slow scan mode; when it finds a beacon that operates in a zone observed by Arahub, the scanning mode is changed to fast. Now, the beacons are scanned with a 1 second period. The RSSI values from all beacons, that are registered to a particular zone, are read and immediately send to the data aggregation system. When a particular beacon from the list is not in range, then such information is also noted. After a long period without any signal form a known beacon, the application switches back to slow scan mode. An alternative version of the application is used in one of the test environments, where the user may also interact with the application to provide his preference related to a product being presented on a display connected to the Arahub system.

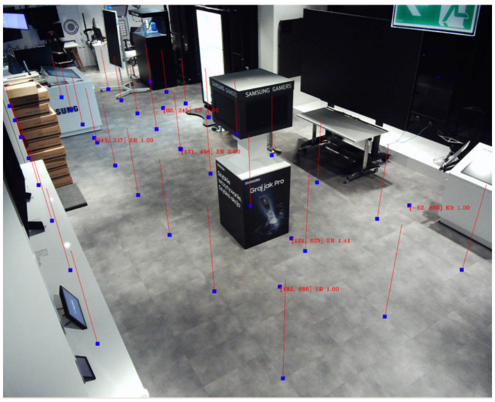

Fig. 3. A view from camera with calibration data shown. A uniform grid of points transformed using the calibration matrix is used to enable human validation of the process.

C. Calibration

In order to obtain physical positions of objects, a calibration procedure is required upon Arabox installation. The calibration is required for the purpose of both the visual and Bluetooth/Wifi positioning systems.

Visual system calibration is done independently for each camera in a particular location. For that, a dedicated chessboard pattern is used with the addition of several markers. The procedure requires placing the pattern and markers in the field of view of the camera - covering possibly the largest surface. Then the coordinates of markers and chessboard are provided to a particular Arabox configuration using a dedicated calibration tool, obtaining world-to-image-plane point correspondences. Using the point correspondences, a projection transformation from 3D world coordinates to the image plane can be calculated. In our work we assume the pinhole camera model. Thus, in order to perform camera calibration, we estimate both intrinsic and extrinsic parameter matrices along with radial and tangential distortion coefficients. We use the calibration method proposed in [25] implemented in the OpenCV [26] library. An example calibration result is presented in figure 3.

The camera calibration procedure is followed by an offline stage of creating a training data set for the purpose of Beacon/Wifi positioning systems. For this, the operator of the

Arahub system needs to use the mobile application to gather data about RSSI levels from BLE beacons in relation to his position predicted by the visual tracking system. Simultaneously, the WiFi signal strength is also recorded using Arabox WiFi monitors. To achieve the best results, the whole observed area should be covered multiple times.

Due to the possibility of errors or security concerns, some areas visible by the video tracking system needs to be excluded (e.g. areas "behind" mirrors). This is done as the last part of the calibration process.

IV. Data sources and processing

A. Location and height calculation

To improve the accuracy of location and height estimation we calculate, using the camera projection matrix, a line orthogonal to floor surface such that on the image, within some margin, it fits in the detected bounding box. To be considered a good prediction, this person candidate's height has to fit in a possible range. Moreover, the location has to be in an acceptable area defined by the union of convex polygons in spot configuration.

In real-world scenarios, especially in commercial zones, we find a number of objects partially covering customers (occlusion) - and it may not be possible or cost-effective to cover some areas with cameras without dealing with such obstacles. The most common scenario is people partially hidden by store shelves, desks or tables, with the upper body detected by the network and legs invisible, which significantly affects location predictions, especially when the camera angle is highly acute. However, if someone goes behind such an obstacle which cuts off the lower part of the box we are able to detect it because Intersection over Union (IoU) of successive boxes should be within the acceptable threshold, but location difference drastically increases and following three 2-dimensional points should approximately form a straight line: camera location (without height), expected location in current time and new location extracted from the cut-off box. Afterward, if we assume that the head is visible within the box and we know the height of this person, we can draw a line in 3-dimensional location coordinate space, such that it satisfies the following four assumptions forming a linear equation system: its length is equal to the height, projection of its start on camera image is equal to head location within the box, it is orthogonal to the ground and ends there.

B. Human tracking based on video data

Within one Jetson device, there are four stages of processing, each performed using separate thread:

1) *Reading frames from camera*
2) *Human detection* is performed using SSD mobilenet lite [27], fine-tuned on spot-specific data set labeled by full-size SSD, created using recordings from each camera.
3) *Box tracking* integrates detected boxes from each frame into a set of currently tracked persons. Firstly, similarity matrix between each box and person is calculated, then one-to-one assignment is performed [28] based on SciPy [29] implementation. Basing on the score used for this matching, *reliability* of each person is altered - ones that were not matched to anything receive a most severe drop, but if they were previously matched, they will still be able to survive several frames before they disappear. A new person with low *reliability* is created when unmatched box probability exceeds the given threshold. The Similarity between box and person is calculated as a weighted sum of: Intersection over Union of the proposed box with estimated person box in current time (calculated using velocity and previous boxes averaged with momentum), spot location difference and height difference.

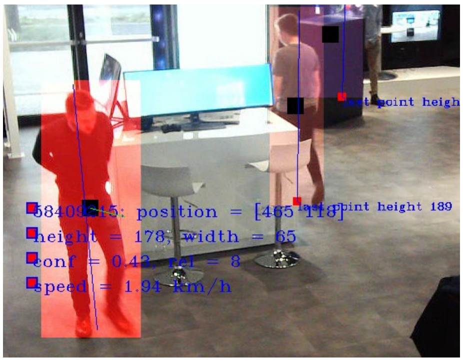

Fig. 4. Detecting real location of partially visible person (man on the right). Since his legs are mostly invisible on the picture, neural network detected only torso. Algorithm detected it and found an approximate point of his feet using head position and height.

4) *Sending* locations and cropped frontal images to server

As a result, the algorithm works with a stable speed of about 8 FPS.

As an alternative to the previous method, when it is possible to place the camera on the ceiling, we propose a tracking approach based on motion detection. This is suitable especially on narrow or crowded passages, where it is hard for people not to cover each other, looking from the side camera.

The first step is image processing to get points that will later be used for clustering. To initially remove noise we use manually implemented Sobel edge detector due to its fast computation on GPU. Afterward, for motion detection, instead of subtracting subsequent frames or saved background image, we use subtracting background computed as the average of previous frames with momentum. With the right parameters, this approach is both resistant to temporarily motionless people (contrary to subtracting subsequent frames) and changing environment i.e. in the form of objects left on the ground (contrary to subtracting saved background). Finally, we choose pixels meeting the given threshold and remove isolated ones that gives us noiseless image.

We tried multiple clustering algorithms using scikit-learn library [30], including hierarchical, OPTICS, Birtch, DBSCAN, K-means and a combination of the last two, however each failed to suit the task. DBSCAN was the closest match, but failed to separate people walking literally side by side. The need was for an algorithm that does not know the number of clusters, is fast with many points (not necessarily many clusters), with the only assumption about the distribution that clusters are denser in the middle, where clusters can touch with a local structure comparable to some clusters interior, however having approximately constant, circular size. Therefore we propose a simple custom approach to clustering based on these assumptions, with the only important parameter being the radius of the cluster and computational complexity $\mathcal{O}(n^2)$, also benefiting from distributed vectorized operations. We calculate the distance matrix between each pair of points, then check for each distance if it is smaller then radius, creating a connectivity matrix for a graph. Then, we iterate over vertices by descending degree and greedily assign a new cluster to check if it does not intersect with any previous (contain vertex already assigned to the cluster). Note that we want that greed because it fulfills the assumption that cluster centers are local maxima of density and without it, if we rewrite the problem into maximizing the number of non-intersecting clusters, two persons side by side are sometimes clustered as three.

To track these clusters, we use the same algorithm as with a neural network based approach, however, instead of IoU of boxes, the similarity of clusters is calculated as symmetric Kullback Leibler divergence, assuming that points form 2-dimensional normal distribution.

C. Merger - connecting the same person's paths from different cameras

To track a person for a longer period of time, we need to merge paths of the same person from different cameras. This is especially desirable in the context of person-device matching, since the longer the path we have, the easier it is to distinguish whether a person has a given device.

The state of the merger algorithm can be represented as a graph, where each path is a vertex and each edge represents the possibility of merging two paths. Within this set, when a new location is added to the path, we only need to update all edges connected to the corresponding vertex, performing computation with complexity independent of their length, unless this triggers merging paths. Managing merges of these vertices is handled using fast `Find-Union` algorithm [31]. In order to simplify the calculation and comparison of paths, locations in the paths are linearly interpolated so that the subsequent timestamps match fixed intervals. Note that the path is processed using the Kalman Filter, so it is enriched with information about the variance, interpreted as the certainty of location prediction. There are three events that can happen after receiving a new location:

1) Initialization: Initialization of a new path after receiving an unknown identifier. Assuming the local camera tracker does not already track this person with a different identifier, edges are added to each vertex, except the ones originating from the same camera.

2) Reject: Rejects are removals of edge from the graph. This happens, when corresponding locations (in time, with their variance) from different paths do not pass Two-Sample t-Test for Equal Means [32], so that within a certain confidence interval, we know that these locations do not originate from the same distribution.

3) Merge: Merges have lesser priority then *Rejects*, as we only take into consideration current, not removed edges. Therefore edges of a merged vertex are the intersection of component vertices neighborhood. This is intuitive and helpful because if given two paths were simultaneously tracked on the same camera in the past or separated significantly, we remember that they cannot originate from the same person also after merge with another path. In practice, in most cases we merge vertices connected by only edge left by *Rejects*, however this is not the case, when pair of people walks together tracked with two cameras, always maintaining close distance. When two paths coexist for a given time without *Reject*, similarity of paths X and Y is calculated as $(\|\mathbf{D}(\mathbf{X}, \mathbf{Y})\|_\mathbf{2})^{-\mathbf{1}}$, where $\mathbf{D}(\mathbf{X}, \mathbf{Y})$ is a vector of euclidean distances between corresponding in time path locations. When the value meets the given threshold, the edge is put on *Merge* priority queue with calculated similarity. The queue is resolved each several iterations, maximizing summed similarity of merged edges. Note that in general, it is `MAXIMUM WEIGHTED CLIQUE COVER` problem with weights on edges, which is at least NP-hard (as a generalization of `CLIQUE COVER`). However, since practical instances are generally small and without any complex structures, we found out that greedy heuristic, trying to merge priority queue starting from most similar edges is good enough.

D. Bluetooth / WiFi signal modeling

We propose two methods for WiFi and Bluetooth signal modeling based on Received Signal Strength Indication (RSSI). The first method is a parametric approach based on the Log-distance path loss model. The second approach is a novel non-parametric method similar to the existing probabilistic fingerprinting-based methods.

Since the received signal power generally decreases as the distance between the receiver and the transmitter increases, it is a valid source of information about the current location of the device of interest. However, RSSI values are heavily dependent on the surrounding environment and other factors such as the relative position of the device or the line of sight between the transmitting and receiving devices. Therefore, in both methods, we adopt a probabilistic approach to explicitly model the aforementioned uncertainty, where we are interested in the likelihood of observing an RSSI value conditioned on a current device location. It is important to note that the roles of the transmitter and the receiver in our models are switched when modeling WiFi and Bluetooth signals. For WiFi signals, we model the RSSI at one of our APs that is being transmitted from the person's device. Here, we know the position of the receiving AP, but the location of the transmitting mobile device is unknown. On the other hand, in case of Bluetooth, we model the RSSI value at the mobile device that is being transmitted from one of the BLE beacons. This way, we know the location

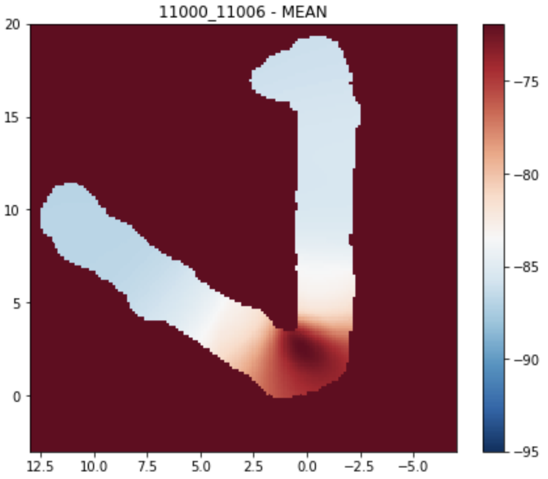

Fig. 5. Heatmap of estimated expected values of the RSSI distribution based on our non-parametric fingerprinting method.

of the transmitting beacon, but the location of the receiving device remains unknown. Another key difference in WiFi and Bluetooth modeling is the fact that in the case of the WiFi the transmitting power of the mobile device is unknown and can vary in time, whereas the transmitting power of the BLE beacon is known and does not change in time. In this work, however, in both cases, we assume that the transmitting power is constant. Thus, we lose on the quality of our WiFI models at the cost of a unified and more transparent approach.

Log-distance path loss model is a radio propagation model that predicts the loss in the signal strength, measured in decibels (dB), inside a building or densely populated areas over distance. We extend the standard log-distance model with the information about the cosine of the angle between the direction the person is facing and the direction of the AP or BLE beacon of interest. This way we can take into account the loss in the signal strength due to the body occlusion, assuming that the device is located at the front of a person. With a further assumption of homoscedasticity of variance and gaussian errors, the log-distance path loss model is a standard log-linear regression model:

$$f(s|x) = \mathcal{N}(s; \beta + \gamma log(d(x)) + w\cos(\alpha(x)), \sigma^2)$$

where s is the RSSI value, x is the device location, $d(x)$ is the distance between the transmitter and the receiver, $\alpha(x)$ is the above-mentioned angle, γ is the estimated path loss exponent that depends on the environment and σ^2 is the estimated variance based on residuals from the fitted model. The key advantage of this method over the second approach is its generalizability. Once we estimate the path loss exponent for a certain environment, we can reuse the fitted model in a different spot location with similar environmental properties, without the offline stage of model training.

Our second approach is similar to the existing fingerprinting-based methods. Here, we assume that we are given a training data set $\{(x_i, s_i)\}_{i=1}^n$ of locations x_i and corresponding RSSI values s_i that where gathered during the offline stage for each AP/BLE beacon in the spot. This data can be gathered efficiently with the help of the video tracking system described in section IV-B. We define a dense grid of point $G = \{x_{i,j}\}$ locations for which we will estimate locally the distribution of RSSI values. In our experiments, the grid had a size of 100×100 with a resolution of less than 0.5 meters. For each point in the grid x_i, we create the set of its nearest neighbors in a given radius r based on the euclidean distance. We define the reliability of each neighbor x_j using the squared exponential kernel with a fixed length scale l - $w_{i,j} = \exp -\frac{\|x_i - x_j|_2^2}{2l^2}$. Next we define unbiased weighted estimators for the mean and variance using the computed reliability weights:

$$\hat{\mu_i} = \frac{1}{V_1}\sum_j w_{i,j} s_j$$

$$\hat{s_i^2} = \frac{1}{V_1 - (V_2/V_1)}\sum_j w_{i,j}(s_j - \hat{\mu_i})^2$$

where $V1 = \sum_j w_{i,j}$ and $V_2 = \sum_j w_{i,j}^2$. Finally, the likelihood of observing a given RSSI value s for a new location x is estimated using the gaussian model with mean and variance of the closest grid point $x_i = \underset{x_j}{\text{argmin}}\|x_j - x|_2$

$$f(s|x) = \mathcal{N}(s; \hat{\mu_i}, \hat{s_i^2})$$

Alternatively, when the spot area is substantially larger and the corresponding grid resolution is lower we can perform linear interpolation of the computed first and second moment estimators prior to likelihood calculation.

E. Human tracking based on radio data

Equipped with a probabilistic signal model we can efficiently tackle the problem of device localization and tracking using either WiFi or Bluetooth signal. We again adopt a probabilistic view of position estimation, i.e. we are interested in computing:

$$x^*_{1:n} = \underset{x_{1:n}}{\text{argmax}}\; p(x_{1:n}|s_{1:n})$$

where each s_i is a set of RSSI measurements observed in a given time window and x^*_i is the estimated location. For notational brevity, we do not distinguish between the AP that received the signal or the transmitting BLE beacon, assuming that for each device we use the corresponding model.

Firstly we focus on estimating position for a single time window. Putting a uniform prior on location $\pi(x) \propto 1$ we calculate

$$p(x|s) \propto f(s|x)\pi(x) = f(s|x) = \Pi_i f(s_i|x)$$

where s_i is a single RSSI measurement. Therefore as the most probable location we simply take $x^* = \underset{x}{\text{argmax}}\; \Pi_i f(s_i|x)$.

To account for spatio-temporal correlations in device localization we use a first-order Kalman Filter, where the underlying noise process models the acceleration of the tracked object.

As a result, for each time step, we obtain the estimated mean and variance of the device position as well as its velocity.

F. Person - device matching

Person - device matching is a key component of the Arahub system, as it enables combining the information extracted from visual cues, e.g. using face recognition systems, with a rich user history based on the advertising identifier or MAC address. We distinguish two tasks for the person - device matching. *Local matching* is focused on correctly assigning a device, from a pool of visible devices, to the user at the moment of entering a spot of interest, e.g. a LED panel. *Global matching* is a continuous process of performing global assignments of all visible devices to all persons currently tracked within a single spot.

Irrespective of the matching task being performed, we first focus on processing video tracking data together with the incoming signal data. To minimize the computation overhead when performing local matching, the process of combining the information about the location of a person at a given time with the incoming signal value is performed in an online fashion. We match a readout about the location with a given RSSI value if their corresponding time difference is less than a specified threshold, usually half a second. When a new signal readout from a device is received, we try to match it with all currently visible tracks. Similarly, when a new location readout is received, we try to match it with all active devices. After successfully matching a location x to a signal value s, the likelihood $f(s|x)$ is computed using one of the models described in section IV-D. The matching system also handles track merges, by taking the union of the location readouts for each track and computing new location-to-signal matches if necessary. Moreover, to provide stable performance over time, we clean up information about inactive tracks and devices.

To solve the *Local matching* task we once again refer to the probabilistic approach. Assigning a device to a person can be formulated as taking a device with the highest conditional probability of observing its signal conditioned on a given track $f(s^i_{1:n_i}|x_{1:m})$. However, to account for a varying number of received signal readout for each device n_i, we focus on maximizing the geometric mean of the total likelihood instead:

$$s = \operatorname*{argmax}_{s^i} f(s^i_{1:n_i}|x_{1:m})^{1/n_i}$$

To solve the *global matching* problem, we first define a cost matrix C, where each entry $c_{i,j}$ represents the cost of assigning a device i to a person j and is equal to the average log-likelihood of observing a total signal s^i conditioned on the tracking locations x^j. We assume independence between each device signal readouts, conditioned on the location, obtaining

$$C = [c_{i,j}]_{i,j} = \frac{1}{n_i} \sum_k \log(f(s^i_k|x^j_{1:m}))$$

Finally, we solve the linear assignment problem [28] using the matrix C to obtain person-device matching. In both local and global matching, if the resulting average log-likelihood of observing a given device signal conditioned on a track is lower than a predefined threshold, we omit this pair in the final assignment.

V. Evaluation

To provide automated testing for algorithms and adjust parameters, we created a simple video tagging procedure. We define convex polygons covering locations space and count for every person where it started and ended its walk and compare its path with manually annotated. This is suitable for both tracking methods. Also using this procedure, we can count how many people entered some room or provide statistical information on people flow around different areas in the commercial area or even shelves.

To reliably test the difficult cases of counting people entering and leaving the room using the motion-based camera mounted on the ceiling, we created a test at a hallway with three exits. In each pass, two people walked touching shoulder to shoulder and either diverted, or walked close together to one exit. The metric was, as described above, how many people passed between each pair of areas, creating a total of 28 manually tagged passes. The algorithm achieved an accuracy around 0.93. The test was carried out in this way because for an analogous, non-directed test in which people naturally and independently entered rooms with 35 passes, the effectiveness was errorless.

A. Use-cases and applications

In order to test the Arahub system in real-life scenarios, we installed it in two sites, that were similar to our target installation environments. Both sites were closed, private spaces yet a substantial number of different people were moving around, thus we could test the system without having control over the environment and people involved.

1) Office Lab: The first location was placed in an office space, that included about 30 persons. Arahub system was installed along a L-shaped corridor that connected all offices, conference rooms, reception, kitchen and utility rooms. The map of the location is presented in figure 6. In total, we used 5 Araboxes - two in each branch of the corridor and one on the bend. They were placed such that it was possible to observe a person entering through the main entrance in the reception and then moving along the corridor, passing all the offices and rooms till the end of the office space. Moreover, two digital displays were installed in the corridor: one at the entrance near the reception desk and the second one at the end of the corridor near a bathroom. In addition, 7 BLE beacons were installed in the corridor in order to uniformly cover it with BLE signal.

The office lab was used for our initial tests and tuning of the system. Our goal was to enable the following minimum requirements for the system:

1) track continuously three persons moving together with spacing between them not less than 3m.
2) track continuously three smartphones that have our custom application installed and running

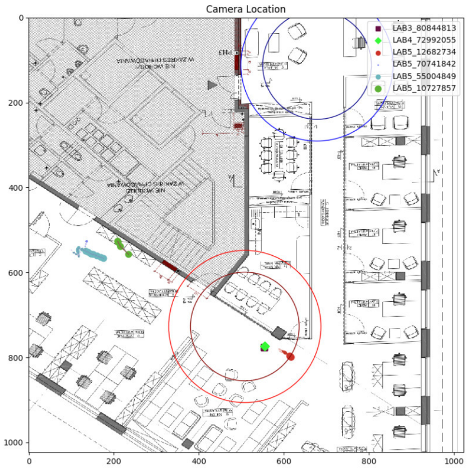

Fig. 6. A map of the office lab. The circles indicate places in which the Arahub system performed an action when human was present - in this case the information about that person was shown on a digital display.

3) be able to assign a smartphone to a person when it approaches a digital display, with accuracy of 80%, after each person walked the distance of the whole corridor length.

Eventually, Arahub system was able to perform according to those three requirements. However, the final accuracy depended heavily on the type of smartphones used in the test. Android-based devices were tracked accurately in about 70% of cases, while for iOS-based devices the accuracy was over 90%. The accuracy was calculated based on 30 trials - separately for both types of devices.

2) Showroom Lab: The second test location was placed in a showroom of one of our business partners. The showroom is a space dedicated to presenting new products to customers; it consists of a large hall with different displays on the walls and conference room. Arahub was deployed to cover the main hall were customers were guided by the showroom's employee. In total 7 Araboxes were installed in addition to 10 BLE beacons. Moreover, one extra Arabox was placed on the ceiling in a narrow part of the showroom - it was used for testing the person counting functionality. The showroom was occupied by 1-2 employees all the time and several times a day, a group consisting of up to 8 people was guided by them. Two digital displays already installed in the showroom were used for the needs of Arahub. Moreover, an alternative mobile application was created - in this version the user could choose one of three products in the application, then a video material, related to this product, was played as this person moved near one of the selected displays.

B. Experiments

The showroom lab was used for testing the performance of Arahub's person tracking capabilities (without person re-identification). In the test, the lab was divided into three sub-areas observed by seven araboxes with overlapping fields of view: 1) narrow corridor - visible by 2 araboxes, 2) large hall with multiple obstacles - visible by 4 araboxes, 3) small hall with one obstacle - visible by 3 araboxes. The goal was to continuously track persons moving between sub-areas. We performed tests in which from 2 to 4 persons were moving across the whole lab using different paths. Moving between sub-areas was counted as a transition. If the system was not able to track a person during a transition, it was counted as a tracking error. Additionally two cases were tested: persons moving separately (not touching each other) and persons moving jointly (without visible separation between them). The results are presented in table I. We may conclude that the arahub system is able to track separately moving persons with high accuracy. However, as re-identification functions were not used, it had difficulties to track persons moving in very close proximity.

TABLE I
EVALUATION RESULTS - CONTINUOUS TRACKING OF PERSONS MOVING BETWEEN PREDEFINED LOCATIONS IN AN AREA OBSERVED BY 7 ARABOXES WITHOUT PERSON RE-IDENTIFICATION

test no.	case	transitions	transition errors	number of persons	accuracy
1	joined	4	3	2	0,25
2	joined	8	3	2	0,63
3	joined	9	3	2	0,67
4	separated	4	1	4	0,75
5	separated	4	0	1	1,00
6	separated	9	0	2	1,00
7	separated	11	1	2	0,91

VI. Conclusions and Future Work

In this work, we have provided a comprehensive description of the Arahub system. We have shown that it is possible to successfully integrate tracking data from video system and smartphones and use it for commercial purposes. Our work was tested in real-life environments, and however it is still at an advanced prototype level, we are able to deploy it in commercial applications. In our work we developed several novel methods for improving tracking and integration of multi-modal signals, we also focused heavily on optimization to provide a solution that is cost-efficient.

The Arahub system needs to be developed towards more versatile usage capabilities e.g. in outdoor environments, or for high-density crowd scenarios. Moreover, the biggest issues are connected to incompatibility between different smartphone brands and systems. Our tests show that even covering 80% of the smartphone brands currently available on the market, requires a substantial amount of fine-tuning. In order to scale the system, a more granular approach of data analysis could be introduced, e.g. person tracking could be done at the crowd level initially, but at a single-person level when more details are needed [33], [34], [35].

We are also developing methods for improving privacy concerns. The current version of Arahub is meant to be deployed in controlled environments, where users have may opt-in freely. There is a need to provide anonymization methods [36],

[37], which would ensure that even the system operator is not able to use the system for other means than statistical analysis of visitors. We are researching the possibility of using novel cryptography methods, that allows one to use data for machine learning purposes without revealing private information.

Acknowledgment

Let us thank Wojciech Rosiński for his great contribution to this work. We also appreciate the support offered by Jetline and Samsung who provided testing space for our experiments.

References

[1] R. Frączek, B. Cyganek, and K. Wiatr, "Parallelized algorithms for finding similar images and object recognition," *Computer Science*, vol. 14, no. 1, 2013.

[2] M. Meina, A. Janusz, K. Rykaczewski, D. Ślęzak, B. Celmer, and A. Krasuski, "Tagging firefighter activities at the emergency scene: Summary of aaia'15 data mining competition at knowledge pit," in *FedCSIS 2015*, 2015, pp. 367–373.

[3] J. Wilson, S. Chaudhury, and B. Lall, "Clustering short temporal behaviour sequences for customer segmentation using LDA," *Expert Syst. J. Knowl. Eng.*, vol. 35, no. 3, 2018.

[4] I. Rüb, M. Matraszek, P. Konorski, M. Perycz, A. Waśniowski, D. Batorski, and K. Iwanicki, "30 sensors to mars: Toward distributed support systems for astronauts in space habitats," in *ICDCS 2019*, 2019, pp. 1704–1714.

[5] J. Bułat, K. Duda, M. Duplaga, R. Frączek, A. Skalski, M. Socha, P. Turcza, and T. P. Zieliński, "Data processing tasks in wireless gi endoscopy: Image-based capsule localization navigation and video compression," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 2815–2818.

[6] H. Lu and M. A. Cheema, "Indoor data management," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016, pp. 1414–1417.

[7] J. Domaszewicz, S. Lalis, A. Pruszkowski, M. Koutsoubelias, T. Tajmajer, N. Grigoropoulos, M. Nati, and A. Gluhak, "Soft actuation: Smart home and office with human-in-the-loop," *IEEE Pervasive Computing*, vol. 15, no. 1, pp. 48–56, 2016.

[8] A. Krasuski, A. Jankowski, A. Skowron, and D. Ślęzak, "From sensory data to decision making: A perspective on supporting a fire commander," in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 3, 2013, pp. 229–236.

[9] D. H. Hepting, R. Spring, and D. Ślęzak, "A rough set exploration of facial similarity judgements," in *Transactions on Rough Sets XIV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 81–99.

[10] M. Świechowski and D. Ślęzak, "Introducing logdl - log description language for insights from complex data," in *FedCSIS*, 2020.

[11] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[12] Q. Dong and W. Dargie, "Evaluation of the reliability of rssi for indoor localization," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*. IEEE, 2012, pp. 1–6.

[13] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal processing magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[14] X. Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila, "Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems," in *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000.*, vol. 6. IEEE, 2000, pp. 3015–3020.

[15] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *2006 3rd annual IEEE communications society on sensor and ad hoc communications and networks*, vol. 1. Ieee, 2006, pp. 374–382.

[16] A. Zhang, Y. Yuan, Q. Wu, S. Zhu, and J. Deng, "Wireless localization based on rssi fingerprint feature vector," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 528747, 2015.

[17] A. Golovan, A. A. Panyov, V. V. Kosyanchuk, and A. S. Smirnov, "Efficient localization using different mean offset models in gaussian processes," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014, pp. 365–374.

[18] B. F. D. Hähnel and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceeding of robotics: science and systems*. Citeseer, 2006.

[19] L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, and Y. Chen, "Using inquiry-based bluetooth rssi probability distributions for indoor positioning," *Journal of Global Positioning Systems*, vol. 9, no. 2, pp. 122–130, 2010.

[20] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does btle measure up against wifi? a comparison of indoor location performance," in *European Wireless 2014; 20th European Wireless Conference*. VDE, 2014, pp. 1–6.

[21] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[22] J. S. Kulchandani and K. J. Dangarwala, "Moving object detection: Review of recent research trends," in *2015 International Conference on Pervasive Computing (ICPC)*, 2015, pp. 1–5.

[23] R. L. F. Fleuret, J. Berclaz and P. Fua, "Multi-camera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, p. 267–282, 2008.

[24] R. Iguernaissi, D. Merad, K. Aziz, and P. Drap, "People tracking in multi-camera systems: a review," *Multimedia Tools and Applications*, vol. 78, 09 2018.

[25] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[26] G. Bradski, "The opencv library. dr. dobb's journal of software tools," 2000.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, p. 21–37, 2016.

[28] D. F. Crouse, "On implementing 2d rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[29] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[30] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] B. A. Galler and M. J. Fisher, "An improved equivalence algorithm," *Commun. ACM*, vol. 7, no. 5, p. 301–303, May 1964.

[32] D. H. Jones, "Book review: Statistical methods, 8th edition george w. snedecor and william g. cochran ames: Iowa state university press, 1989. xix + 491 pp," *Journal of Educational and Behavioral Statistics*, vol. 19, no. 3, pp. 304–307, 1994.

[33] W. Świeboda, A. Krauze, and H. S. Nguyen, "A granular evacuation modeling framework," *Annals of Computer Science and Information Systems*, vol. 2, p. 337–342, 2014.

[34] M. Świechowski and D. Ślęzak, "Granular games in real-time environment," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 462–469.

[35] M. Przyborowski, T. Tajmajer, L. Grad, A. Janusz, P. Biczyk, and D. Ślęzak, "Toward machine learning on granulated data – a case of compact autoencoder-based representations of satellite images," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 2657–2662.

[36] P. Szczuko, "Simple gait parameterization and 3d animation for anonymous visual monitoring based on augmented reality," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10 561–10 581, Sep 2016.

[37] B. Cyganek, "Change detection in multidimensional data streams with efficient tensor subspace model," in *Hybrid Artificial Intelligent Systems*. Cham: Springer International Publishing, 2018, pp. 694–705.