

# RealTrac technology at the EvAAL-2013 competition

Alex Moschevikin <sup>a,\*</sup>, Aleksandr Galov <sup>a</sup>, Alexander Volkov <sup>a</sup>, Alexander Mikov <sup>a</sup>, Sergey Reginya <sup>a</sup>, Roman Voronov <sup>a</sup>, Oleg Reut <sup>a</sup>, Maria Serezhina <sup>a</sup>, Alexey Zaitsev <sup>a</sup>, Pavel Lunkov <sup>a</sup>, Sergey Malodushev <sup>a</sup>, Dmitry Kirienko <sup>a</sup>, Alexander Fedorov <sup>a</sup>, Alexey Sementsov <sup>a</sup>, Sergey Podryadchikov <sup>a</sup>, Konstantin Spiridonov <sup>a</sup>, Ivan Tershukov <sup>a</sup>, Artem Yushev <sup>a</sup>, Alexander Nuikin <sup>a</sup>, Kirill Gostev <sup>a</sup>, Sergey Pashinsky <sup>a</sup>, and Alexei Soloviev <sup>a</sup>

<sup>a</sup> *RTL-Service JSC and Petrozavodsk State University, Lenin Str., 31, 185910 Petrozavodsk, Russian Federation*  
E-mail: alexmou@mail.ru

**Abstract.** In this paper, the results of the EvAAL-2013 localization competition are examined. A comprehensive description of RealTrac™ technology, the winner of this competition, is presented. Focused on real-time location technology, RealTrac hardware includes a server, access points (gateways and repeaters), and mobile devices (voice intercoms and tags). A location calculation engine is based on Bayesian algorithms fusing data from time-of-flight and received signal strength measurements, inertial measurement units, building structure, and optionally, from context events. The peculiarity of the embedded navigation system is that it works effectively independent of the place where the mobile tag is actually attached, i.e., it could be mounted on one's foot, carried freely in a jacket pocket, held in one's hand, or even hung on a shoestring. The RealTrac system can easily be combined with third-party systems via the Real Time Location System Communication Protocol that consists of an open API and uses a common Keyhole Markup Language format for geo-data presentation. Furthermore, in this paper, it is shown that the scores obtained in the competition regarding positioning accuracy might be significantly higher when the applied particle filter is appropriately configured and a probability-based approach for area-of-interest determination is introduced.

**Keywords:** Local positioning system, indoor navigation, Bayesian filter, time-of-flight, received signal strength

## 1. Introduction

A rise in the application of real-time location (RTL) technology has intrigued the public sector and research agencies, with both exploring new paths of innovative communication. In addition to usage by the general public, social scientists are exploring opportunities for utilizing RTL. By monitoring, collecting, linking, and interpreting RTL data, a new form of social gathering has emerged. One of the major transitions in recent research is the increased requirements for positioning accuracy, motivating a variety of new research programs in and across various competitions.

### 1.1. Social dimension of RTL

Demographic trends of aging populations all over the world have raised the importance of issues involving support and care of the elderly. The duty of new generations is to provide opportunities for such individuals to continue living a safe, rewarding, and interesting life. These opportunities should include, but not be limited to, medical treatment and convenient living conditions. In this new era, a combination of information and communication technologies can help us achieve these goals.

Ambient assisted living (AAL) [1] addresses these goals by exploiting new technologies and exploring ways in which they can be integrated into the everyday lives of senior citizens as well as people in hospitals and nursing homes. AAL provides solutions for

---

\* Corresponding author. E-mail: alexmou@mail.ru

creating safe and easy-to-use environments, in particular navigation assistance inside buildings and communication with personnel.

Numerous technologies have been researched and applied to provide location information indoors. Typical positioning systems include the following three components: equipment, techniques, and location estimation algorithms [2]. Equipment can be further classified as infrastructure (e.g., beacons, base stations, anchors, and transceivers) and mobile devices (e.g., receivers and tags). Note that the mobile device is usually attached to one's body. The entire set of techniques can further be classified by a signal used for position estimation; this includes radio signals (i.e., radio frequency), near field communication and radio frequency identification (RFID), and infrared or ultrasound waves. Some techniques do not require the use of mobile devices, for example, video surveillance or ultrasonic systems, which measure changes in the monitored field of view.

For wireless radio signals, it is possible to measure not only the received signal strength (RSS) but also the time-of-flight (ToF) value and angle of arrival. The location algorithm must fuse measurements from different sensors, apply some constraints, and produce a final estimate of the object's position. Variations of Kalman or particle filters are usually employed for this task [3].

The EvAAL competition [4] aims at establishing benchmarks and evaluation methods for comparing AAL solutions. The primary goals of the contest are to identify leading suitable technologies, define how these could be implemented within ambient living environments, and suggest how to transfer them into the industry.

### 1.2. Indoor localization competition

In July 2013, teams from France, Portugal, Italy, and the Russian Federation, as well as three teams from Germany, participated in the indoor localization competition held in Madrid, Spain. In addition to our RealTrac team, the description of the systems used by other participants is as follows:

- The LOCOSmotion system used WLAN fingerprinting and accelerometer-based dead-reckoning [5].
- The AmbiTrack system provided marker-free localization and tracking via a camera-based approach [6].
- FEMTO-ST/HMPS presented a system coupling outdoor positioning service with Wi-Fi based fingerprinting and trilateration system, as well as a marker analysis system [7].
- The IPNlas indoor localization system was based on fingerprinting in Wi-Fi network using an Android smartphone [8].
- The Magsys localization system measured (via a wearable receiver) the strength of the magnetic field formed by stationary transmitters. Next, these measures were transformed into position and orientation information. The underlying principle was the registration of resonant oscillating magnetic fields [9,10].
- The AALocation localization system [11] used the ZigBee network and was based on the fusion of data from RSS measurements and presence sensors [12].

Our positioning system, presented by the Russian RealTrac team, earned the first prize in the localization track. Some aspects concerning RealTrac technology were published in [13]. The purpose of our paper is threefold. First, RealTrac technology, particularly concerning AAL systems, is described. Second, the algorithms applied during EvAAL-2013 are justified. Third, possible improvements to our achieved results by tuning the localization algorithms are discussed. These settings were determined based on a posteriori analysis of the competition results.

The remainder of the paper is organized as follows. Section 2 contains a description of the EvAAL contest, including competition rules and procedures. In Section 3, a comprehensive overview of RealTrac technology, including network architecture, hardware, protocols, the RealTrac API, voice communication, and other details is provided; note that in this section, the necessary aspects for integrating RealTrac and AAL systems are discussed as well. Section 4 is dedicated to the developed positioning technique, disclosing details regarding the particle filter, the principles of the applied inertial measurement subsystem, and the justification of the algorithm applied for area-of-interest (AoI) determination. In Section 5, the obtained scores are summarized, and in Section 6, the results regarding point-to-point localization accuracy and the success of AoI determination are evaluated. Methods for possible improvements in both these aspects are demonstrated. Furthermore, particle filter performance and dependence on the number of particles is discussed. Finally, Section 7 is devoted to conclusions and outlines future goals.

## 2. EvAAL-2013 competition

Evaluating AAL systems through competitive benchmarking (EvAAL) is an activity focused on evaluating solutions related to AAL by organizing annual competitions. The goal of these competitions is to compare different AAL solutions and provide developers and researchers with an arena in which their technologies can be tested. Evaluation methods and benchmarks are also subject to ongoing refinement and improvement. In addition to comparing algorithms, EvAAL focuses on such solution parameters as cost, deployment difficulty, interoperability, among others.

The third EvAAL competition was organized at the Smart House Living Lab of the Polytechnic University of Madrid, Spain; more specifically, this is where the indoor localization track and AAL tracking was deployed.

The Living Lab building is a single-story building with a control room, a bathroom, and common space comprising a kitchen, a living room, and a bedroom. It also has a porch with a ceiling. In the competition, the localizable space includes the bathroom, common space, and porch.

The floor plan of the Living Lab is presented in Figure 1, with the height of the ceiling varying from 2.32 to 2.62 m. The Living Lab has some home appliances (e.g., television set, refrigerator, oven, and washing machine). Inside the Living Lab, sensors and actuators are placed for different purposes, for example, flood sensors in the kitchen and bathroom, smoke and fire sensors next to the kitchen, and magnetic contact sensors on some doors and windows. A domotic bus provides notifications for all events, with each event recorded with its specific coordinates. This information is available for use by a localization system.

### 2.1. Competition rules and procedures

Although localization systems have actively been investigated and used for a long time, there is no common standard as to how to develop them, and which services and components they should have. Therefore, the competition's organizing committee proposed a set of criteria to comprehensively rate systems not only with respect to accuracy but also convenience for daily use, ease of integration into house interior, installation complexity, and reliability.

These five criteria were weighted according to their importance as follows:

- Accuracy (weight 0.35),

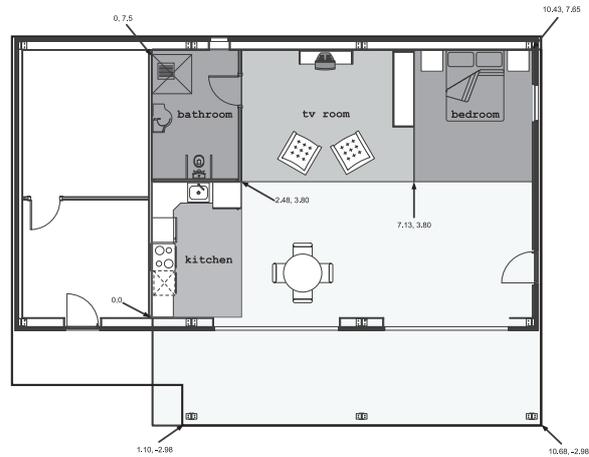


Fig. 1. Smart House Living Lab plan

- Availability (0.20),
- Installation complexity (0.10),
- User acceptance (0.20), and
- Interoperability with AAL systems (0.15).

Total scores were calculated by weighting all five criteria with the corresponding weight factors.

The accuracy of each system was evaluated by averaging the scores across the three phases described below.

In the first phase, the positioning system should determine the AoI in which the object is. The typical AoI included the bathroom, the front of the kitchen, and the area near the dining room table.

In the second phase, the positioning system should estimate the real-time position of a single user in the scene without any disturbances.

In the third phase, the system should evaluate the user's position in the presence of another person who moves freely inside the localization area.

The AoI accuracy score was given by the expression  $10 \cdot T + 5 \cdot U$ , where

- $T$  is the fraction of time in which the localization system provides the correct information regarding user presence in a given AoI and user absence from any AoI;
- $U$  is the fraction of time during which the system confuses a small area with a big area that contains a small area (big areas are presented in Figure 1; small areas are not shown).

To calculate the point-to-point accuracy score, the individual error of each measurement was evaluated as the Euclidean distance between the measured and ex-

pected points. Next, the 75th percentile  $P$  of the errors was determined and scaled according to the formula below.

$$\text{Score} = \begin{cases} 10 & \text{if } P \leq 0.5m \\ 12 - 4 \cdot P & \text{if } 0.5m < P \leq 2m \\ 8 - 2 \cdot P & \text{if } 2m < P \leq 4m \\ 0 & \text{if } P > 4m \end{cases} \quad (1)$$

The organizers allowed the competitors to install their equipment within 1 h. The time spent preparing was incorporated into installation complexity criteria (see rules in [14]).

By the end of the installation stage, trajectory recording began. If the technology assumed the use of mobile tags, then such tags were given to the actor in advance. In some cases, the actor was instructed to wear the tag in a specific manner. For our RealTrac system, the actor was free to place the tag anywhere on the body since the technology does not have any restrictions regarding tag placement on a user's body—it can be in one's pocket, in one's hand, or even hanging on a shoestring as a necklace. Next, the actor moved along a few predefined paths with stops at predefined points, as shown in Figure 2. The true ground trajectory was not disclosed to competitors.



Fig. 2. An actor moves along a predefined trajectory; our mobile tag was placed freely in the left front trouser pocket

The phases indicated above were performed twice, and the best marks were recorded. After completing this portion, the last two evaluation criteria (i.e., user acceptance and interoperability with AAL systems) were evaluated using questionnaires. Many questions regarding every aspect of the developed system were asked—from device size and charging requirements to the ability to utilize the system in the Open Source community.

## 2.2. Installation at the Living Lab

Previous research showed that typical errors of ToF ranging for nanoLOC hardware indoors were usually comparable with the size of the room [15]. Therefore, given an area of 121 m<sup>2</sup>, only four access points were used as anchors in the Living Lab. The anchors were installed near the corners of the room to cover the entire localization area. Two access points were placed at metal beams near the ceiling of the porch area ( $h \approx 2.5$  m) and two between the light plates of the ceiling and roof in the bathroom and bedroom. Positions of anchors were marked by numbers 1–4, as shown in Figure 3. The access point marked with as 5 was used as a gateway; it relayed data to and from the wireless segment.

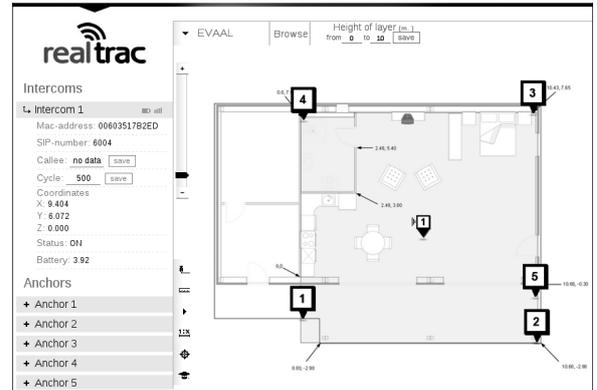


Fig. 3. RealTrac graphical web interface with numeric marks indicating the placement of access points in the Living Lab

Since all access points were powered from external batteries, the installation process was very quick—the installer spent only 2 min and 1 s on system deployment.

RealTrac localization algorithms used the following three data sources: ToF data, RSS data, and trajectory data recorded by an embedded inertial measurement unit (IMU). Localization accuracy depends on the quality of these three data sources. Among these three sources, the IMU data contributed to the accuracy to the greatest extent.

Given this, it was decided to hide all access points in the ceiling. On the one hand, this decision led to less-accurate ToF ranging; on the other hand, it resulted in better user acceptance criteria scores.

The use of additional anchors inside the room was not reasonable. It did not lead to any increase in location accuracy because the NLOS errors were high and

an additional ToF radius did not yield a substantial reduction in the localization area.

Note that the organizers provided access to all location-based events that occurred in the system in the Living Lab. For example, manually switching the lights on or off generated the corresponding event. In this case, the localization server could position the person who pressed the button near the switch. Unfortunately, in our 2013 RealTrac system, the ability to use third party geo-events in the localization algorithms was not implemented by authors.

Detailed results of the RealTrac system are presented in Section 5. Final scores and the jury decision scores of all competitors are summarized in Table 2.

### 3. RealTrac technology description

RealTrac technology has been developed by RTL-Service JSC (Petrozavodsk State University, Russian Federation) since 2008. Applicable to the localization of mobile devices and voice communication between intercoms, such technology is based on the nanoLOC radio standard (IEEE 802.15.4a CSS) developed by Nanotron Technologies GmbH. The main features of this standard include a high data rate of 1 Mbit/s, an automatic ToF round trip time ranging procedure, and low power consumption.

#### 3.1. Network operation scheme

As shown in Figure 4, the RealTrac system comprises mobile devices (intercoms and tags), access points, and a server.

Access points form a radio segment. They can measure distances to mobile devices using the ToF method and send these data to the server. Access points are connected to the server via Ethernet; therefore, a RealTrac system can be installed in a building with an existing wired network.

The server estimates mobile device location after receiving the measured distances from the access points. Locations can be displayed on a browser on any laptop connected to the Internet. Furthermore, a third-party application obtaining location data from the server can be developed, for example, through HTTP.

Radio segment coverage can be enlarged using access points operating in a repeater mode.

Depending on the application, the RealTrac system can support any number of mobile devices.

The effective time of one frame transmission, including interframe space, headers, and an acknowledgement, is  $\approx 1.4$  ms. In the round trip time (RTT) scheme, each ToF measurement consists of three successive Data/Ack frame exchanges. Therefore, the total time required for one RTT ranging is  $\approx 5$  ms.

To determine the tag inside the localization area, at least three ToF measurements from the access points are required. In practice, four or five such measurements are recommended. Thus, the system is not permitted to perform more than 20–25 measurements/s, as the 2.4 GHz ISM bandwidth is limited.

Based on the above, the maximum number of mobile devices in one network segment depends on the distance measurement frequency. For example, if the position update period is 1 s, then the network can process no more than 20 mobile tags. If the position update period is 10 s, the number of tags can be increased to 20, and so on.

For technical details on the RealTrac radio system and operation algorithms, see [13]. Additionally, the subsections below describe the main characteristics of the hardware devices and software components of the system.

#### 3.2. RealTrac hardware units

The RealTrac system consists of different types of hardware units. All the hardware devices are roughly divided into four groups, as mentioned below.

1. Mobile intercoms and tags (Figure 5a-5c)
2. Access points –data transfer devices providing radio network infrastructure (Figure 5e-5f)
3. An accessory device for primary network setup and testing performance (Figure 5d)
4. A Server –a node that collects and analyzes data and manages the sensor network

##### 3.2.1. Mobile devices

Mobile devices (intercoms and tags) are equipped with the following:

- pressure sensor,
- temperature sensor,
- RSS meter,
- three-axis accelerometer,
- three-axis gyroscope,
- three-axis magnetometer, and
- RFID tag.

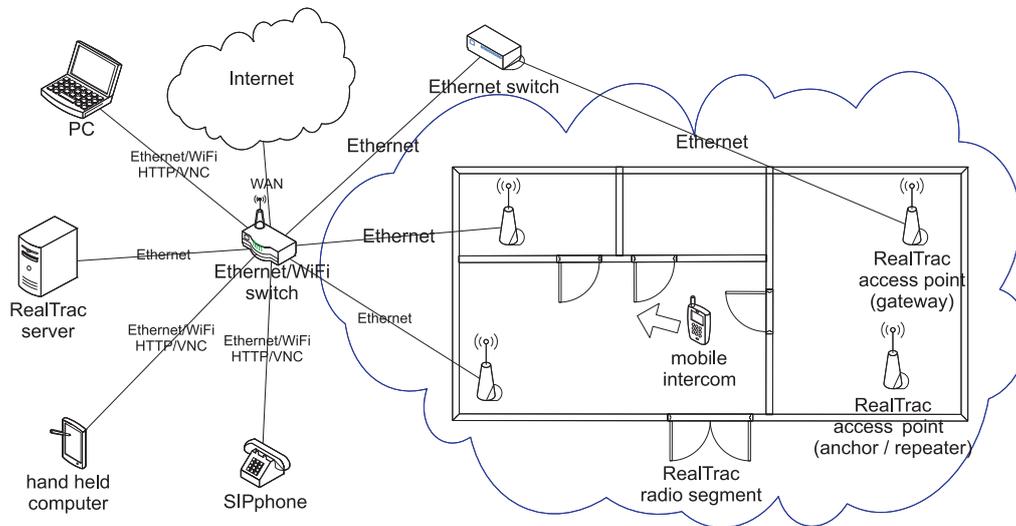


Fig. 4. RealTrac technology network diagram



Fig. 5. RealTrac technology hardware: (a) duplex intercom Rio®; (b) tag –mobile sensor; (c) half-duplex intercom radio Sentry®; (d) wireless handheld network analyzer; (e) industrial outdoor access point; and (f) indoor access point

An intercom operates in a cyclic manner at a certain frequency –i.e., it wakes up, listens to its radio segment for incoming packets, performs certain operations (if requested by the server), then sleeps during the rest of its cycle time. Thus, it provides the required

frequency of distance measurements while conserving battery power. The duration of this cycle can be adjusted by server-side software.

When an intercom accepts an incoming call or initiates a call itself, it is put into the *sound* mode and does not sleep at all. After the call is finished, the intercom switches back to the power-saving mode. A tag operates according to the same scheme.

### 3.2.2. Access points

Stationary RealTrac nodes (i.e., access points) are equipped with the following:

- pressure sensor,
- temperature sensor, and
- RSS meter.

The two key features of the access point are distance measurement and data transfer from the radio segment to the server.

The access point operates in one of two modes, either *gateway* or *repeater* mode. In the *gateway* mode, the access point receives data from mobile nodes including ranging results, system data, voice streams, and sensor data. It then transfers these data to the server.

The *repeater* mode implies that the access point repeats all received packets from the wireless network and increments the *hops* field in doing so. Evidently, it skips packets sent from itself and only retransmits packets that it has never repeated before.

Access points participate in a distance ranging procedure in both modes.

The mode is automatically selected when the device starts up. If the access point is connected to Ethernet when the power is turned on, it operates in the *gateway* mode; otherwise, it operates in the *repeater* mode. In the *gateway* mode, the access point sends a broadcast DHCP request to obtain an IP address from a local DHCP server.

An access point communicates with the RealTrac server through the Inter Nano Communication Protocol (INCP) [13].

If a local network contains a specifically preconfigured DHCP server, then this server returns the IP address of the INCPd server that the access point should work with. Next, the access point communicates with the INCPd server directly using INCP.

Otherwise (i.e., if a local network contains only a common DHCP server), the access point obtains the default network settings and sends a special DNS request to which a DNS server returns the IP address of the INCPd server that serves the given access point.

In the *repeater* mode, the access points do not have IP addresses and hence wireless INCP communication goes over the nanoLOC data link level.

The effective bandwidth of gateways and repeaters is limited by the radio throughput and does not exceed 500 Kbps.

The total occupied bandwidth of the network depends on the number of installed repeaters. Use of repeaters decreases the performance because of frame duplication, unnecessary retransmissions, and unavoidable collisions.

Note that there is only one case in which using a large number of repeaters is possible, namely, in mines, where long chain-like segments would be organized.

### 3.2.3. Analyzer

The analyzer is an essential device for system maintenance and deployment assistance. It can be used for experiments, radio monitoring, and other tasks. Apart from other RealTrac hardware, the analyzer is equipped with a keyboard and graphical display for enhanced user interaction.

Features provided by an analyzer include the following:

- searching and identifying nearby active devices;
- displaying information regarding devices, filtered on-demand;
- displaying information regarding network status, such as RSS and number of neighbors; and
- configuring analyzer settings.

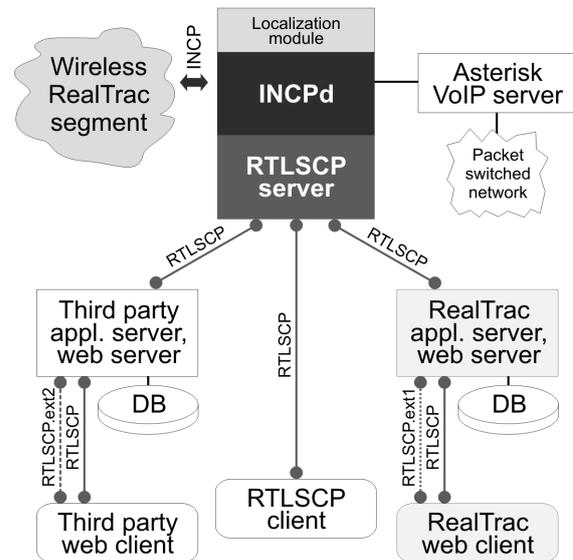


Fig. 6. Modular software structure of the RealTrac technology

The analyzer is equipped with an 1800 mAh battery, which guarantees that the device can work continuously for up to 14 h without recharging.

### 3.2.4. Server

For the EvAAL-2013 competition, a Dell Inspiron Ultrabook with 4 GB RAM and an Intel® Core™ i5-3317U (1.7 GHz) processor was used as the server.

### 3.3. Software architecture

As illustrated in Figure 6, the modular software structure of the RealTrac technology comprises the following components:

- RealTrac server (localization module, INCPd server, and RTLSCP server);
- Asterisk voice communication server;
- Optional application server (see "RealTrac application server" and "Third party application server" in Figure 6); and
- Client software (see "RealTrac web client" and "Third party web client" in Figure 6).

The INCPd server is responsible for the intercommunication between the hardware and software components. This server manages the wireless network, integrates the access points into the system via Ethernet, and mediates between hardware devices and other software components. As a result, software components can communicate with hardware devices as with other software entities. The INCPd server and access points

use INCP to provide data and sound transfer to and from them. A brief description of INCP is presented in [13].

One of the key components of the INCPd server is a localization module that calculates the location of a given device using all available information. The localization module uses its own database for storing temporary data required for further location estimation.

The INCPd server handles the following data regarding devices:

- global coordinates,
- movement trajectory obtained from an embedded IMU,
- number of human steps obtained from IMU,
- device battery voltage,
- position update cycle frequency,
- measured absolute pressure,
- measured temperature, and
- measured RSS value of an incoming radio frame.

The RealTrac server provides an API by implementing the Real Time Location System Communication Protocol (RTLSCP) (see "RTLSCP server" in Figure 6). This public API is a part of the RealTrac API. On the basis of the representational state transfer (REST) architectural style, RTLSCP uses JSON notation for requests and responses, and was designed to simplify third-party application development (see "Third-party application server" component connected to "RTLSCP server" in Figure 6).

The basic RTLSCP version provides only a few features, including device list and location requests and device configuration options. Location information is provided in global coordinates (i.e., latitude, longitude, and altitude) that works well with common geoservices such as Google Maps. RTLSCP is described in more detail in Section 3.5.

In addition to the basic RTLSCP version, the extended version of the protocol consists of optional features such as location history, event notifications, and voice calls history. The RealTrac system incorporates the extended RTLSCP version as a separate component (see "RealTrac application server" in Figure 6 and note the second connection with "RealTrac web client" marked as "RTLSCP.ext1" that reflects the extended RTLSCP version).

Technically, the application server is an RTLSCP client of the RealTrac server. It might obtain, for instance, location data from the server and store such data in a separate database to provide location history on demand. Data caching might be implemented on the

client side, but a better place to realize such business logic is on a server.

The application server provides the following information through the extended RTLSCP:

- location history,
- maps images storage,
- mobile device owner details,
- distance from a given point to a device and its direction,
- event notifications related to entrance to and exit from a given area, as well as two devices meeting,
- event notifications related to RFID-based access control,
- object's motion details (e.g., static, walking, running, and cycling), and
- alert notifications (e.g., fall of the owner and alert button click).

The INCPd and application server are both developed using JDK 7 and various open source libraries.

A third-party developer can extend the application server to provide additional data to a client via new JavaScript modules; however, the provided RealTrac application server and its web client implement most common features and are ready to use as is. The web client application is a good example of how to integrate the RealTrac system into a corporate application (or an AAL system).

### 3.4. Voice communication

Voice communication is provided by the open source PBX Asterisk software (see "Asterisk VoIP server" in Figure 6). The RealTrac system supports duplex and half-duplex voice communication between intercoms and between an intercom and a static SIP-phone connected to the server. The system can be extended to support voice communication between RealTrac hardware and traditional telephones or cellular phones by installing a special telephony adapter on the server.

When intercoms are moving within the wireless network coverage, automatic roaming is supported by the system server. In such cases, sound packets are sent through the nearest access point to the given intercom.

Voice communication is the most resource-intensive feature of the entire system. Accordingly, there are quite evident requirements for the server hardware.

The RealTrac voice communication feature is described in more detail in [13].

### 3.5. RealTrac API

As localization in AAL scenarios can be exploited by other applications to offer advanced services, integrability becomes a very significant parameter for localization systems.

Modern software and hardware systems are often complex and continuously evolve during the maintenance phase. The architecture of such systems must be robust to changes to enable support and keep the development process easy and effective. Furthermore, the ability to extend such systems is even more important.

A stable system architecture has well-defined interfaces between core components of the system and public APIs for third-party extensions and use. Special attention should be paid to public system APIs because their usability causes the success (or failure) of the system among developers, and consequently, in the market.

Common principles of effective system APIs include the following:

*Simplicity.* The API must be easy to learn and use. Ideally, the API does not require extra documentation to learn; a few descriptive paragraphs should be sufficient to start.

*Implementation independence.* Implementation independence allows developers, for example, to change the programming language used for the implementation of the API without changing the API itself. This also means that third-party developers can use any language or technology to create clients of the API.

*Robust to change.* A good API must be robust to changes in the underlying technology. For example, new hardware might be substituted for old, and the API of the system based on the old hardware should not require changes.

*Extensibility.* API extensibility allows core developers to add new features to the system without negatively impacting published API versions.

The RealTrac system is equipped with REST-based HTTP services that comply with all of the above principles. These HTTP services form the RTLSCP –the public API for the RealTrac system. Formats of requests and responses of these REST-based services use JSON notation. The principles of a good API listed above are implemented as follows:

- Simplicity is provided via the JSON notation itself. JSON is easy for individuals to read and write. Moreover, RTLSCP uses well-known terminology such as *anchor*, *beacon*, and *node* in re-

quest specifications. Hence, no special documentation is required for understanding the protocol.

- To achieve implementation independence, the current RTLSCP implementation is based on the Java Virtual Machine; however, it could be changed without affecting RTLSCP, because JSON is completely language independent and does not limit core and third-party developers as CORB or other RPC technologies and notations do.
- For robustness, if the hardware of the RealTrac technology changes, then RTLSCP is still applicable because hardware does not touch the items of RTL.
- Extensibility is provided by the REST specification since any number of new services could be created by simply adding new URLs and corresponding request specifications to the list of available RTLSCP services.

RTLSCP covers most common features of any RTLS and provides services for handling RealTrac hardware and visualization. The common API provides the following data:

- Real-time locations of anchors and mobile nodes,
- Lists of anchors and mobile nodes, and
- Location calculation results in absolute and relative coordinates.

The following additional data structures are served by the RealTrac APIs:

- Anchors and mobile nodes parameters and statuses, and
- Keyhole Markup Language (KML) files for visualization.

An example of a typical RTLSCP request and corresponding response are presented in Listing 1 and Listing 2, respectively. The request contains one parameter, a list of MAC addresses whose locations are requested.

Listing 1: RTLSCP request example 1

```
{ "request": {
  "mac": [
    "BAD90234FF41",
    "BAD90234FF42"
  ]
}}
```

The response consists of the list of structures that in turn consist of the MAC address, the location calcu-

lation time, and the coordinates of the object. In this example, the (x, y, z) coordinates are relative to some known pivot.

Listing 2: RTLSCP response example 1

```
{ "response": [
  {
    "mac": "BAD90234FF41",
    "unixtime": "1343291851",
    "coord": {
      "loc": {
        "x": 32.7,
        "y": 45.3,
        "z": 0.0
      }
    }
  },
  ...
]}
```

One of the main RTLS options is to visualize the calculated location in relation to specified physical objects. The easiest way to present a location in a web browser is via Google Maps. Given the growth in the popularity of web interfaces in recent years due to their simplicity of implementation and support, as well as cross-platform features [16], KML-formatted files were proposed for use in RTLSCP. KML is natively supported by both Google Maps and Google Earth applications. A distinct REST service in RTLSCP allows all data for visualization to be acquired as one KML file; more specifically, this includes anchors and mobile node locations and statuses.

Listing 3 gives an example of such a KML file; it consists of coordinates and metadata for *walls* and *anchors*, and defines the color of wall lines. First, the building name ("ITpark / ITpark.floor.2") and wall line color ("FF0000FF" in <alpha><RGB> format) are defined. Second, wall coordinates are presented, and finally, the list of anchors and its properties are defined.

Listing 3: RTLSCP response example 2

```
<?xml version="1.0" encoding="UTF-8"
  standalone="yes"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:gx="http://www.google.com/kml/ext
/2.2" xmlns:xal="urn:oasis:names:tc:ciq:
xsdschema:xAL:2.0">
<Document id="ITpark.floor.2">
  <name>ITpark / ITpark.floor.2</name>
  <Style id="style.wall">
    <LineStyle>
      <color>FF0000FF</color>
      <width>1.0</width>
```

```
</LineStyle>
</Style>
...
<Folder>
  <name>walls</name>
  ...
  <Placemark>
    <styleUrl>#style.wall</styleUrl>
    <MultiGeometry>
      <LineString>
        <altitudeMode>absolute</
          altitudeMode>
        <coordinates>
          34.35426029047551,61.78706865853971,77.0
          34.35353384612551,61.78683938835208,77.0
          ...
          34.35367522330444,61.78686074498395,77.0
        </coordinates>
      </LineString>
      ...
    </MultiGeometry>
  </Placemark>
</Folder>
<Folder>
  <name>anchors</name>
  ...
  <Placemark>
    <name>0000CAFE0001</name>
    <styleUrl>#style.anchor</styleUrl>
    <ExtendedData>
      <Data name="rtls.object">
        <value>anchor</value>
      </Data>
      <Data name="rtls.anchor.mac">
        <value>0000CAFE0001</value>
      </Data>
    </ExtendedData>
    <Point>
      <altitudeMode>absolute</altitudeMode>
      <coordinates>
        34.35355700038095,61.78683500209754,77.0
      </coordinates>
    </Point>
  </Placemark>
  ...
</Folder>
</Document>
</kml>
```

The REST service allows one to request actual object locations as another KML file, which provides only new location coordinates that can be used to update existing positions on the map. Using a similar approach, it can also provide updated anchors positions.

As an example, to incorporate this KML file into a Google Earth application, it is first necessary to use a special KML file that describes a *resource* to obtain actual locations (see Listing 4). The structure and format

of the file with new locations is almost identical to the main KML file; therefore, it is not presented here.

#### Listing 4: RTLSCP response example 3

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>ITpark</name>
    <NetworkLink>
      <name>locations</name>
      <Link>
        <href>http://127.0.0.1:9002/rtlscp/
          location/ITpark/kml/</href>
        <refreshMode>onInterval</refreshMode>
        <refreshInterval>1</refreshInterval>
        <viewRefreshMode>onRegion</
          viewRefreshMode>
      </Link>
    </NetworkLink>
    ...
  </Document>
</kml>
```

Third-party developers can easily integrate these two KML files into standard Google Earth applications while testing the RealTrac system during deployment.

All location coordinates are received in one of the following two formats:

- absolute (latitude, longitude, altitude);
- relative (x, y, height).

By default, KML files contain locations in absolute coordinates. Two location coordinate formats allow third-party developers to begin with a simple relative coordinate format that can be quickly integrated into lightweight RealTrac client applications. Rich web and desktop applications that use Google Maps plugins or other plan visualization utilities that require latitude, longitude, and sometimes altitude can benefit from the absolute coordinates format.

In summary, RTLSCP provides a simple and robust public API that can be used to implement additional business logic, thus providing the possibility of integrating RealTrac technology into existing corporate and open source systems.

#### 3.6. Integration of RealTrac system and AAL

Integration with AAL software was one of the key parameters in the overall scoring criteria at the EvAAL-2013 competition. The RealTrac system mainly conforms with the integrability criteria defined by the

organizers, since the technology provides an API to RTLSCP, as described in Section 3.5.

To integrate the RealTrac software system with the EvAAL benchmarking software, the so-called "light" integration procedure was used [17]. This procedure consisted of sending simple text messages to the benchmarking software over TCP. The developed middleware requested device coordinates from the RealTrac server software using RTLSCP commands, and then sent localization data to the EvAAL benchmarking software with a 2 Hz update frequency. The same middleware generated AoI events.

The text message with localization data contained the following fields:

- competitor identifier (a string);
- estimated X coordinate of the object, represented in decimal form;
- estimated Y coordinate of the object;
- integer representing the POSIX timestamp of the current location estimation; and
- list of numbers representing the AoIs in which the object is located.

## 4. Positioning technique overview

The RealTrac technology is based on nanoLOC (IEEE 802.15.4a) transceivers operating in a 2.4 GHz bandwidth. This radio standard utilizes the chirp spread spectrum technique and provides automatic ToF RTT ranging.

Ultra-wideband (UWB) technologies (also defined within the IEEE 802.15.4a standard) are the main competitors to nanoLOC in terms of ToF measurements. UWB has a substantial advantage in that it provides more accurate distance measurements since the spectrum is considerably wider; however, the range of reliable communication is less. Regulations on radio transmission power set an upper limit of -41.3 dBm/MHz on a signal's power spectral density emission for UWB transmitters.

The communication range is tightly connected to the spatial density of the deployed equipment and influences the overall costs of the local positioning system. This enables nanoLOC-based hardware to technically compete in some market niches, particularly in applications that require location accuracy of 1–3 m.

As mentioned above, the most accurate and least costly positioning might be achieved by an inertial navigation system. With this approach, the trajectory

is obtained by an IMU embedded in a majority of off-the-shelf handheld devices. In addition to this trajectory being relative to some origin, the use of these data is a powerful technique for increasing localization accuracy. The data from IMU can be used separately to determine the indoor location [18,19] or fused with information from wireless sensor networks for better location accuracy.

Numerous systems utilize IMU data together with RSS data. Such systems are based on RFID [20] technologies or WiFi [21,22]. The latter is widely used because of the ease of implementation on smartphone platforms. Another group of systems uses IMU together with the UWB technique [23,24,25]. Though there are several systems processing IMU data together with RSS or UWB measurements, there is currently a lack of systems that use IMU together with ToF RTT nanoLOC data.

The use of IMU as an additional source of information for nanoLOC-based systems allows such systems to compete with UWB-based systems in terms of installation complexity and system costs, as well as with RSS-based systems in terms of localization accuracy.

The choice of which positioning algorithm should be used depends on the information available. A majority of IMU-plus-RSS-based systems use particle filters for data fusion. The IMU-plus-UWB-based systems use the extended Kalman filter for processing more accurate UWB measurements.

Furthermore, the Kalman filter can be used for fusing ToF RTT-based measurements in the 2.4 GHz range with IMU data [26]. However, this approach requires accurate IMU data that can be measured by a wheel-based robot and absence of NLOS errors that can be achieved only outdoors. The positioning system focused on indoor pedestrian localization implies less accurate IMU data. The presence of severe NLOS errors in ToF measurements indoors makes the application of the Kalman filter considerably more difficult.

The RealTrac technology enables measurements of both distance and path loss between an access point and a mobile node. The IMU data are used to improve positioning accuracy. The localization server is configured to use KML-formatted information about the locations of access points and building structure (i.e., walls, stairs, rooms, corridors, and furniture). As depicted in Figure 6, location calculation starts as the IN-CPd server resends the following data to the localization module: MAC addresses of access points, MAC address of a mobile device, corresponding set of dis-

tance and path loss measurements, and information from the embedded IMU.

Further details concerning IMU algorithms are discussed in Section 4.1.

Several positioning algorithms that fuse this information have previously been studied [15,27]. For the EvAAL competition, the particle filter was chosen to calculate the position of the object (see the description in Section 4.2). This novel approach for the fusion of IMU data with ToF data allows us to construct the particle filter without using the weights.

#### 4.1. IMU-based localization

Positioning accuracy might be significantly improved if the three-axis accelerometer and gyroscope are used. The developed IMU-based navigation engine does not restrict the typical usage scenarios of a RealTrac device and assumes it could be placed wherever the person finds it suitable.

To increase the battery lifetime, the power-saving mode of operation was implemented. While the accelerometer is always in an active state, the gyroscope is turned on only when motion or a walking pattern is detected. The gyroscope calibration settings are calculated during normal operation and stored in non-volatile memory.

The IMU-based engine includes distance, orientation, and trajectory calculation modules, described in detail in [28].

The first module estimates the total distance travelled by counting the number of steps and calculating step lengths. The step counter detects a step with an accuracy that exceeds 98%. The proposed step counting technique showed advanced performance without regard to orientation of the device, walking speed, or physical characteristics of the person. For step length  $L_{step}$  estimation, the following formula was identified and used as the most efficient for normal walking speed [28]:

$$L_{step} = K \cdot t_{step} \cdot \sqrt[4]{a_{max} - a_{min}}, \quad (2)$$

where  $t_{step}$  is the duration of the step and  $a_{max}$  and  $a_{min}$  are the maximum and minimum vertical acceleration values during the step time, respectively. Note that accelerometer samples were filtered by a moving average, the length of which depends on sampling frequency and lies in the interval [8, 20]. Optimal calibration constant  $K$  was obtained by processing the experimental data gathered from ten subjects.

Quaternions were used to represent the altitude of the device. Because of the limited computational resources of the microcontroller unit (MCU), the complementary filter in time discrete form was applied [29]. A person's heading was extracted from quaternion observations as the mean of two two-dimensional vectors corresponding to the last two steps.

The position update was performed on the step basis. As the system detects a step, it acquires a step length and a step heading direction. Both are passed to the trajectory calculation module in which a new position point is added to the trajectory. All processing is performed by the internal MCU of a mobile device. As the time to send the data to the server approaches, new calculated trajectory points along with ToF and path loss data are included in the INCP message.

IMU performance was evaluated through experiments. The Return Position Error parameter, i.e., the computed return position in comparison with the real one, for multiple tests did not exceed 6% of the total travelled distance [28].

The IMU trajectory can provide position estimation even if ToF measurements are unavailable, and can be used as an extra constraint in particle or Kalman filtering techniques.

#### 4.2. Particle filter

To fuse all information available in the localization server, the particle filter was used. The particle filter is a very efficient algorithm that is widely used for location calculation with noisy measurements [30]. To represent the uncertainty of an object position, a set of random samples or particles with corresponding weights is used. Each particle represents the possible state of the mobile object, considering information regarding object position and direction. Particles move according to the selected motion model and are subject to constructional restrictions; for example, they cannot pass through walls. After several steps of the algorithm, the particle set converges to the factual state of a mobile object.

In the current implementation, at each time moment  $t$ , the system state is characterized by the set of particles  $X_t^{(i)}, i = 1..N$ , where each particle  $X_t^{(i)}$  corresponds to the possible position  $(x_t, y_t)$  and orientation  $\theta_n$  of the located object. Each particle moves according to the trajectory acquired from the IMU; therefore, it is not necessary to introduce information regarding target velocity into the motion model.

For simplicity, the ToF measurements are used only for geometrically removing particles from the location area. Therefore, no weights are used. The typical phases of the particle filter algorithm, including initialization, propagation, update, resampling, and state estimation, are described below.

##### 4.2.1. Initialization

During the initialization phase, the set of  $N$  particles is randomly generated within the area of the intersection of circles corresponding to the ToF measured distances, as illustrated in Figure 7. The motion directions of the particles are uniformly distributed in the range from 0 to 360 degrees.

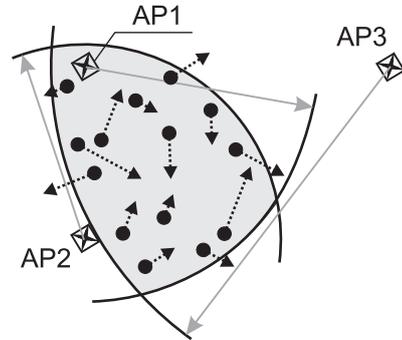


Fig. 7. Initial particles, generated inside the circle intersection area; AP1, AP2, and AP3 are access points, forming the circle intersection area; the radii of the circles are equal to the ToF measured distances; solid dots denote particles; and the arrows indicate particle positions after the propagation phase

##### 4.2.2. Propagation

In the propagation phase, the next position of a particle is determined by the trajectory obtained from the IMU. The trajectory between two sequential moments of ToF measurements  $t - 1$  and  $t$  is presented as the number of vectors  $s_j, j = 1..J$ , as shown in Figure 8. Each vector  $s_j$  includes information about its length  $l_j$  and heading  $\alpha_j$ .

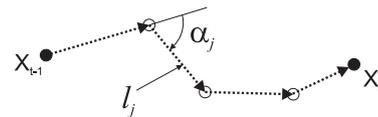


Fig. 8. Example trajectory acquired from an embedded IMU; black circles show the moments of ToF measurements,  $t - 1$  and  $t$ ; white circles correspond to the  $J - 1$  checkpoints set by the IMU

The particle coordinates and direction are changed according to the following motion model equation:

$$x_t = x_{t-1} + \sum_{j=1}^J (l_j + \delta l_j) \cdot \cos \alpha_j, \quad (3)$$

$$y_t = y_{t-1} + \sum_{j=1}^J (l_j + \delta l_j) \cdot \sin \alpha_j, \quad (4)$$

$$\theta_t = \theta_{t-1} + \sum_{j=1}^J (\alpha_j + \delta \alpha_j), \quad (5)$$

where  $(x_t, y_t)$  and  $(x_{t-1}, y_{t-1})$  denote coordinates of the target location at corresponding time moments  $t$  and  $t - 1$ ,  $\theta_t$  and  $\theta_{t-1}$  are the calculated and previous target directions, respectively, and  $l_j$  and  $\alpha_j$  are the step length and direction, respectively.  $\delta l = N(0, \sigma_l)$  is a normally distributed random step noise with deviation  $\sigma_l$  that corresponds to the possible changes in step length of an object, and  $\delta \alpha = R(-\alpha, \alpha)$  is uniformly distributed random noise that corresponds to the possible changes in object direction. The errors  $\delta l$  and  $\delta \alpha$  are empirically set with dependence on the quality of the information acquired from the IMU.

Structural information regarding the building is used to increase positioning accuracy [31]. If a particle crosses a wall during its motion, the particle is removed from the current set (see Figure 9).

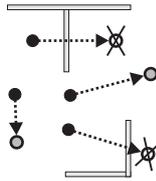


Fig. 9. Particle propagation restricted because of building structure

After the propagation step, only  $K$  out of  $N$  particles survive. It is possible that the final particle set is empty (i.e.,  $K = 0$ ). In this case, the system is re-initialized according to the first step of the algorithm. For example, the re-initialization procedure could occur when the object significantly changes the direction of its motion or the period between measurements is extremely large.

#### 4.2.3. Update

The update procedure is applied when new ToF measurements are received. First, the location area is formed by intersecting the circles corresponding to the ranging results. Second, the propagated particles that occurred outside the intersection are removed from the final set. Thus, after removal, only  $M$  particles survive.

#### 4.2.4. Resampling

Since there is no weight calculation used in the described algorithm, a resampling procedure differs from the typical procedure [30]. Each particle from the set of  $M$  surviving particles is duplicated  $n$  times, where  $n$  is the closest integer to  $\frac{N}{M}$ .

#### 4.2.5. State estimation

As a final step, the object location is calculated as the average of the particles as follows:

$$\hat{X} = \frac{1}{N} \sum_{i=1}^N X^{(i)}. \quad (6)$$

#### 4.2.6. Algorithm summary

The particle filter algorithm is presented below using pseudo-code.

The following input data is used at each iteration of the algorithm:  $\{X_{t-1}\}$  is the set of particles at the previous step,  $\{Z_t\}$  is the set of current distance measurements,  $U_t$  is the trajectory data registered with the IMU, and  $N$  is the initial number of particles.

1. Draw the area of circle intersection corresponding to the received measurements  $\{Z_t\}$  (see Fig. 7).
2. If  $\{X_{t-1}\}$  is empty, generate set  $\{X_t\}$  of  $N$  new particles inside the area of circle intersection and go to step 7.
3. Calculate new particle positions  $\{X_t\}$  by propagating each particle from set  $\{X_{t-1}\}$  with new data  $U_t$  according to (3), (4), and (5).
4. From set  $\{X_t\}$ , remove those particles that intersected a wall during propagation (see Fig. 9).
5. From set  $\{X_t\}$ , remove those particles that are located outside the area of circle intersection  $\{Z_t\}$ .
6. Resample the rest of the particles  $\{X_t\}$  by duplicating  $M$  remaining particles  $n$  times, where  $n$  is the closest integer to  $\frac{N}{M}$ .
7. Calculate object location by averaging the resampled particles from set  $\{X_t\}$  via (6).

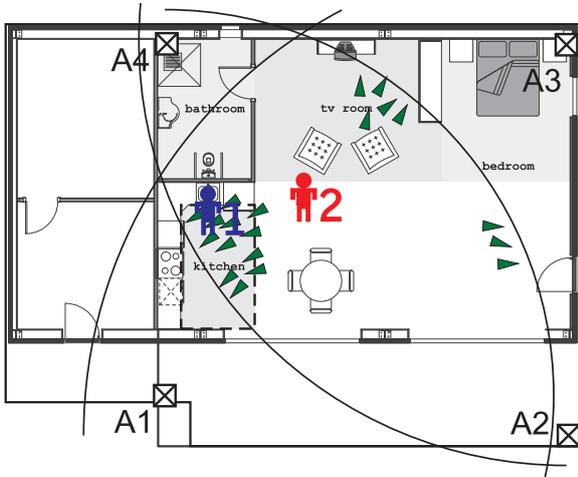


Fig. 10. Area-of-Interest determination by a particle cloud; A1, . . . , A4 are access points with circles corresponding to the ToF measurements; kitchen, bathroom, tv room, and bedroom are AoIs; 1 indicates the true position of the pedestrian; 2 indicates the averaged (by all particles) position of the pedestrian (incorrect proposition); green triangles indicate particles corresponding to the possible locations of the pedestrian

#### 4.3. AoI determination

There are two ways to determine the area within which an object is located. The straightforward approach is to determine whether the location point is inside a certain AoI. This approach was used in the EvAAL-2013 competition.

A more robust approach uses a particle state representation, i.e., based on the probability of an object being in a certain area. This could be estimated as the number of particles inside the area divided by the total number of particles. An event stating that the object is inside the area is generated when the corresponding probability exceeds a certain threshold, for example 50%.

A true-to-life situation happened at the EvAAL-2013 competition and depicted in Figure 10 demonstrates these two approaches.

In the figure, the true position of the pedestrian is labeled 1. Furthermore, the particle cloud is separated into three groups. One group is located near the true position of the pedestrian in the kitchen area. The two other groups are in the tv room and the right bottom corner of the hall near the door. Straightforward calculations according to (6) yield the object's position marked by label 2. Thus, the object belongs neither to the kitchen area nor the tv room, and the corresponding event of being in the certain AoI did not occur.

Since a majority of particles are concentrated in the kitchen area, however, their relative number exceeds the threshold for AoI determination. The AoI event might be generated in this case, a scenario corresponding to the ground truth.

Note that in Section 6, the term "the improved AoI determination algorithm" is used to refer to this probability-based approach.

## 5. Results of the EvAAL-2013 competition

The organizers used two approaches to test the performance of the competing positioning systems. First, they determined point-to-point accuracy for predefined traces. Second, contestants provided real-time data upon entering certain AoIs. To compare the results of the locations, a calculation of the 75% quantile of the positioning error cumulative distribution function (CDF) was used.

There were four different traces with different lengths and shapes. The ground and calculated traces with the corresponding CDF obtained during the EvAAL-2013 competition for the RealTrac positioning system are shown in Figure 11.

In the figure, the solid lines in the left plots correspond to the ground traces, whereas the dashed lines correspond to the traces calculated by the RealTrac system during the EvAAL competition. The OX and OY axes of the left plots correspond to OX and OY coordinates of the building. The CDFs for the corresponding traces are shown in the right plots, with the OX axis corresponding to the point-to-point accuracy (i.e., positioning error) and the OY axis corresponding to the probability of having a certain positioning error. The black arrow indicates the error for the 75% quantile for the given CDF.

For the first (shortest) trace, the 75% quantile of point-to-point accuracy was equal to  $\approx 3$  m, for the second trace, it was  $\approx 2$  m, for the third trace, it was  $\approx 1.9$  m, and for the fourth (longest) trace, it was  $< 1.7$  m. The efficiency of the particle filter depended on the convergence speed of the algorithm. The longer the trace, the better the accuracy results achieved. For the shortest trace (trace 1 in Figure 11), the algorithm did not have sufficient time to converge, thus leading to poor accuracy results ( $\approx 3$  m @75%).

Detailed results obtained by the RealTrac team in comparison with other competitors are presented in the next two tables.

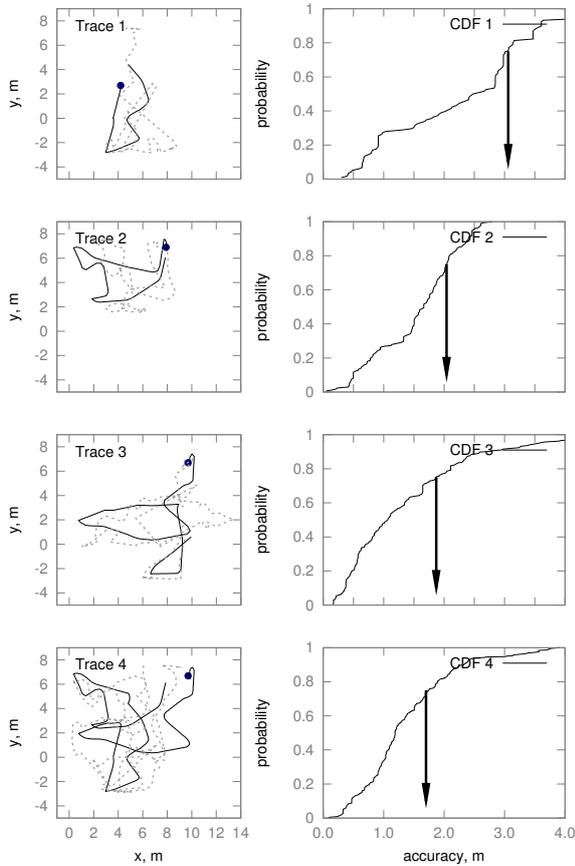


Fig. 11. Four traces and corresponding CDFs during the EvAAL-2013 competition; black solid dots indicate the starting points of the traces; the OX and OY axes of the left plots correspond to coordinates of the building; the OX axis of the right plots corresponds to the point-to-point accuracy, whereas the OY axis corresponds to the probability of locating a person with the given accuracy; solid lines on the left plots correspond to ground traces; dashed lines correspond to calculated traces; arrows in the right plots indicate 75% quantile accuracy

Table 1 describes accuracy and availability results during the four walking sessions.

Rows B1.1, B1.2, B2.1, and B2.2 correspond to traces 1, 2, 3, and 4 in Figure 11, respectively. The first two columns show the 75th percentile of the distance between true and estimated positions. The last four columns reveal the scores obtained by the RealTrac system for accuracy and availability during the first and second attempts. In benchmarks B1.1 and B1.2, only one person walked around the Living Lab with brief pauses following the predefined paths. In benchmark B2.1, one person moved along the predefined path; at the same time, an additional actor moved

freely inside the Living Lab space. In benchmark B2.2, the actor walked along the coupled traces from B1.2 and B2.1. For benchmarks B1.1, B1.2, and B2.1, the point-to-point accuracy was selected as the accuracy score; for trace B2.1, the AoI accuracy was measured.

These results show that the RealTrac system converges to the true position of the actor in time. Some loss in the availability score (see columns for the availability criterion for the two attempts in Table 1) can be explained by the fact that the system provided exactly one sample every half second, and sometimes packets were lost or delayed.

Table 2 describes the obtained scores for all participating teams, showing results for each contest criterion, namely, accuracy, availability, installation complexity, user acceptance, and integrability within AAL. The first three scores were calculated by benchmarking and the last two were calculated by a questionnaire. The total score, obtained by weighting, is also presented in the table. Additional information regarding the evaluation procedure is presented in Section 2.

## 6. Algorithms evaluation

After the EvAAL-2013 competition ended, all obtained results were analyzed. It was found that there were potential improvements to the localization algorithm performance for both point-to-point and AoI tests.

For algorithm evaluation, the set of measurements and corresponding ground truth traces collected during the EvAAL competition was used. The set of experimental data consists of four trajectories; each trajectory consists of the set of ground points with corresponding ToF measurements acquired at each point for a given mobile device from several anchors. Experimental data were obtained for each trace two times (i.e., from the two allowed attempts during the competition). One of the attempts was announced as successful and was considered in the final results, whereas the other attempt was rejected.

The general idea behind finding improved and proper settings for the algorithm was as follows. We proposed to optimize the accuracy of the rejected attempt (i.e., to use it as reference data), and then apply these settings for the particle filter algorithm to the accepted attempt. Next, we proposed comparing results of processing the data from the accepted attempt by the algorithms with the two sets of settings (i.e., the set used at EvAAL-2013 and the optimized set). We

Table 1

RealTrac accuracy and availability results. B1.1 – one person, first path (the shortest one), B1.2 – one person, second path, B2.1 – two people moving around, B2.2 – AoI test

Trace	Accuracy, [m] @75%		Accuracy, scores		Availability	
	Attempt 1	Attempt 2	Attempt 1	Attempt 2	Attempt 1	Attempt 2
B1.1	3.09	<b>3.06</b>	1.8046	1.8787	9.9167	9.8333
B1.2	2.29	<b>2.05</b>	3.4104	3.9006	9.8864	9.7727
B2.1	<b>1.87</b>	2.70	4.5060	2.6088	9.8739	8.5714
B2.2	<b>1.74</b>	2.22	6.8312	6.4961	9.8214	9.7194
Mean	N/A	N/A	4.1381	3.7211	9.8746	9.4742

Table 2

EvAAL-2013 overall scores

Team Name	Accuracy	Availability	Installation complexity	User acceptance	Integrability in AAL	Total
RealTrac	4.14	9.87	10	7.43	8.68	7.21
AmbiTrick	2.46	9.92	7	6.10	9.44	6.18
IPNlas	1.12	10.00	10	7.91	8.03	6.18
LOCOSmotion	3.47	6.59	7	7.53	8.56	6.02
Magsys	1.55	9.24	10	6.66	6.28	5.66
FEMTO-ST	0.52	9.81	10	3.84	8.91	5.25
AALocation	2.15	8.15	0	5.41	4.87	4.20

believe that this approach increases the applicability of our research.

### 6.1. Point-to-point accuracy

The efficiency of the particle filter strongly depends on noise parameters  $\delta l$  and  $\delta \alpha$ , as well as the number of particles. Noise parameters  $\delta l$  and  $\delta \alpha$  depend on the quality of measurements acquired from the embedded IMU. Knowing the characteristics of a person's motion (i.e., step length, and velocity), it becomes possible to adjust them to be optimal for a given person. Furthermore, the number of particles is a critical parameter that influences the convergence rate.

Therefore, we performed a set of simulations to optimize the parameters of noise and the number of particles, and compared the acquired results with the results from the competition.

During the EvAAL competition, the following parameters were used:  $\delta l = N(0, \sigma_l = 0.8)$ ,  $\delta \alpha = R(-\frac{\pi}{4}, \frac{\pi}{4})$ , and number of particles  $N = 200$ . Furthermore, in this paper, these settings are referred to as  $PF_{EvAAL}$ . These settings were determined in the experiments conducted in the IT park of Petrozavodsk State University [27,28] just before EvAAL-2013. Parameters of the IMU noise were chosen to cover data

inaccuracies for all possible cases of holding or carrying the mobile device (e.g., keeping in a pocket or bag, or holding in one's hand). The number of particles was chosen to achieve the best location accuracy during the running of the algorithm on the predefined measurement set.

As was found later (i.e., after EvAAL-2013), the following parameters minimize positioning error for the reference data:  $\delta l = N(0, \sigma_l = 0.3)$ ,  $\delta \alpha = R(-\frac{\pi}{20}, \frac{\pi}{20})$ , and number of particles  $N = 100$ . These optimal settings are referred to as  $PF_{opt}$ .

The particle filter algorithm is based on random particle generation; therefore, the results of the algorithm vary from one run to another. To estimate the average positioning accuracy, it is necessary to conduct a large number of runs for a given set of ground and measurement data.

Results of the positioning accuracy calculated with the particle filter with optimized parameters were compared with those calculated with the particle filter used during the EvAAL-2013 competition. In both cases, a series of 1000 runs of the algorithm for each set of trace data was conducted and processed.

The distribution of accuracy results (75% quantile) acquired within the  $PF_{EvAAL}$  and  $PF_{opt}$  series are

shown in the right plots of Figure 12. In the figure, solid bars show the results of the algorithm used at the EvAAL-2013 competition,  $PF_{EvAAL}$ . Shaded bars show the results of improved algorithm,  $PF_{opt}$ . Arrows on the plots indicate the 75% quantile value obtained at the competition. The left plots show examples of the traces calculated with our improved algorithm. Evidently, the optimized settings of the particle filter yield better results (compare the dashed curves for all traces in Figure 11 and Figure 12).

The mean accuracy of our improved algorithm  $PF_{opt}$  is better as compared with that of  $PF_{EvAAL}$ . The histogram for our improved algorithm is also narrower; the main cause for such a result is the smaller deviation of particle angle noise, which was decreased from  $\delta\alpha = R(-\frac{\pi}{4}, \frac{\pi}{4})$  to  $\delta\alpha = R(-\frac{\pi}{20}, \frac{\pi}{20})$ .

The longer the trace, the narrower the histogram for both algorithms, since for such long traces, the larger part of the trajectory is calculated with the particle filter in the converged state.

## 6.2. AoI determination

During the competition, the straightforward approach was used for AoI determination by the RealTrac localization server. Thus, an AoI event was reported to the AAL system if the object (i.e., the center of mass of all  $N$  particles representing the object) was located within the AoI.

As was noted in Section 4.3, a detailed analysis of the competition results revealed an approach to significantly improve performance of the system for AoI determination. The particle filter allows us to estimate not only the location of an object but also the posterior location density. Furthermore, it makes it possible to use this posterior density for AoI determination by calculating the relative number of particles in all AoIs.

Table 3 presents a comparison of the two approaches of AoI determination for the largest areas in the Living Lab (i.e., bathroom, bedroom, kitchen, and tv room).

Table 3

Percentage of correct AoI determinations for two approaches

AoI	Straightforward, %	Improved, %
bathroom	76	85
bedroom	87	89
kitchen	32	92
tv room	85	93

Since the particle filter is a probabilistic algorithm, results of area determination vary from one run to an-

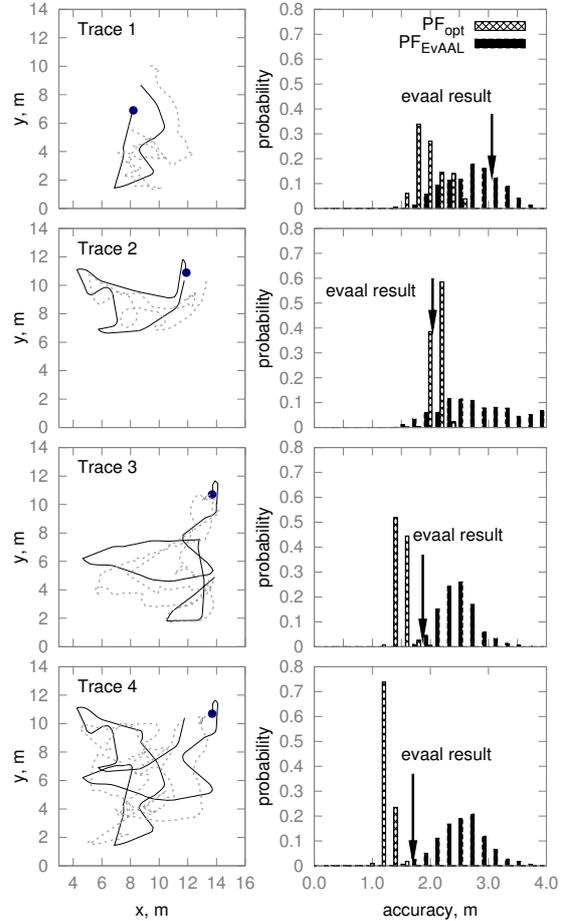


Fig. 12. Examples of traces calculated with our improved algorithm and a comparison of the distribution of accuracy results for both  $PF_{EvAAL}$  and  $PF_{opt}$  with our improved algorithm; in the left plots, black solid dots indicate starting points of the traces, solid lines correspond to ground traces, dashed lines correspond to examples of traces calculated using our improved algorithm ( $PF_{opt}$ ), the point on each trace corresponds to the beginning of the trace, and the OX and OY axes correspond to coordinates of the building; in the right plots, the histograms portray accuracy results obtained over 1000 runs of the algorithms, solid boxes correspond to the algorithms used at the EvAAL-2013 competition ( $PF_{EvAAL}$ ), whereas shaded boxes correspond to our improved algorithm ( $PF_{opt}$ ), the OX axis indicates 75% quantile positioning accuracy, the OY axis indicates the probability of achieving the given accuracy, and black arrows indicate 75% quantile accuracy results demonstrated at the EvAAL-2013 competition

other, much like the position calculation algorithms described in Section 4.2 and discussed in Section 6.1. Therefore, the PDF of the AoI determination results acquired in multiple runs with the same given input data adequately describes the effectiveness of the algorithm.

The results presented in Figure 13 were acquired from 1000 runs for each AoI determination algorithm (i.e., the straightforward and improved approaches).

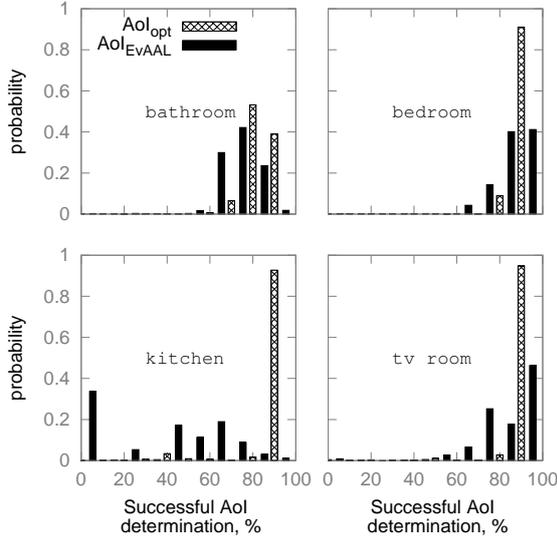


Fig. 13. Distribution of AoI determination results; the OX axis corresponds to the correctness of AoI determination expressed as a percentage; the OY axis corresponds to the probability of the results; solid boxes indicate results generated with the straightforward algorithm used at the EvAAL-2013 competition, whereas shaded boxes correspond to the improved algorithm that uses the particle state representation

In the figure, the OX axis corresponds to the number of successful AoI determinations relative to all attempts. The OY axis corresponds to the estimation of the probability of obtaining the given AoI determination results.

As an example, for the kitchen, the most probable result for the straightforward approach is close to zero, since the area is small and NLOS ToF errors are large. As a result, the system cannot recognize whether the object is within this area. However, using our improved algorithm  $AoI_{opt}$ , the probability of correctly detecting the actor inside the kitchen is equal to  $\approx 90\%$ . Furthermore, the histograms corresponding to our improved approach are narrower, indicating that our improved algorithm shows more stable results from one run to another.

In summary, for the information presented in the plots of Figure 13, the use of the particle representation of posterior location distribution increases the percentage of successful AoI determinations.

### 6.3. Advantages of our improved solution

As mentioned above, all RealTrac equipment and software modules at the EvAAL competition used default general settings defined prior to the event. For example, localization algorithms were tuned to be used without considering the data describing walls and building contours. As another example, the maximum speed of an object was set to a very high limit of 3 m/s, which is not achieved in reality. Additionally, as yet another example, an actor was assumed to put the mobile device anywhere he or she wanted.

In fact, the mobile device was placed (and remained) in the actor's pocket, which significantly decreased IMU inaccuracies. Furthermore, the actor walked with a predefined speed and had a certain step length that allowed a posteriori analysis to optimize step length noise. Data regarding ToF measurements in the Living Lab enabled us to estimate possible location precision and optimize the number of particles. Therefore, we suggested the approach of applying the optimized settings found by analyzing the attempts rejected at the competition (see Section 6).

The use of the optimized parameters of the particle filter  $PF_{opt}$  and  $AoI_{opt}$  significantly improved positioning accuracy and the efficiency of AoI determination. According to Figure 12, the average accuracy was improved by approximately 1 m for all traces; moreover, the percentage of successful AoI determinations for the kitchen increased from 32% to 92% (see Table 3).

### 6.4. Algorithm performance

A key characteristic of the particle filter is the trade-off between the number of particles and processing time. A larger number of particles allows us to estimate location more precisely; however, it requires more computational resources. Figure 14 illustrates the dependence of location accuracy (75% quantile of the CDF) and the average calculation time per location on the basis of the number of particles for the improved versions (both  $PF_{opt}$  and  $AoI_{opt}$ ) of the algorithm. The particle filter ran on a computer with an AMD Athlon II X2 245 2.9 GHz processor using the Ubuntu 10.10 operating system.

As shown in the figure, location error decreases significantly when the number of particles increases from five to approximately 100. For a greater number of particles, location accuracy remains approximately constant, whereas processing time linearly increases.

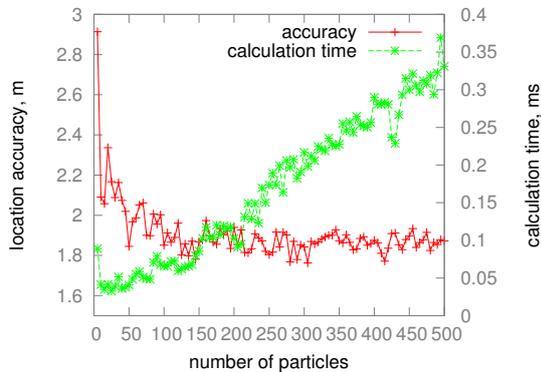


Fig. 14. Accuracy (75% quantile of the CDF) and calculation time per location depending on the number of particles

Therefore, 100 particles were chosen as the optimal setting for the PF implementation.

## 7. Conclusion

The aim of this contribution was to examine results of the EvAAL-2013 indoor localization competition using experiences of the RealTrac team, which was the contest winner. A number of criteria, including positioning accuracy, availability, installation complexity, user acceptance, and integrability within AAL, differentiated our results from previous findings in the literature and the results of other competitors. This differentiation is attributed to the use of a set of special techniques and decisions of the RealTrac system.

In this paper, it was shown that scores obtained at the EvAAL-2013 competition could have been higher if proper localization algorithm settings were applied. Additionally, better results could be achieved by using information from external sensors in the Smart House Living Lab provided by the organizers. For example, location-aware context events regarding actors switching the light on or off, proximity sensors, and the like might have been used for real-time corrections in positioning estimations. Unfortunately, the RealTrac team used no such information in the EvAAL-2013 competition; however, the work after the competition indicates the possibility of further improvements in localization accuracy, which was the highest weighted criterion.

### Acknowledgments

The research described in this paper was ordered by the RTL-Service JSC and was supported by Petrozavodsk State University (within the program of the

Strategic Development Program of PetrSU), the Ministry for Education and Science of the Russian Federation (contract 14.574.21.0059 and other grants), the Foundation for Assistance to Small Innovative Enterprises, and the Ministry for Economic Development of the Republic of Karelia (Russian Federation). The authors would like to thank Enago ([www.enago.com](http://www.enago.com)) for the English language review.

## References

- [1] D. Salvi, P. Barsocchi, M. Arredondo and J. Ramos, EvAAL, Evaluating AAL Systems through Competitive Benchmarking, the Experience of the 1st Competition, *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking*, eds: S. Chessa, S. Knauth, **309** (2012), 14–25.
- [2] I. Amundson and X. Koutsoukos, A survey on localization for mobile wireless sensor networks, *Mobile Entity Localization and Tracking in GPS-less Environments*, 2009, pp. 235–254.
- [3] L. Hui, H. Darabi, P. Banerjee and L. Jing, Survey of Wireless Indoor Positioning Techniques and Systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **37**(6) (2007), 1067–1080.
- [4] P. Barsocchi, S. Chessa, F. Furfari and F. Potorti, Evaluating Ambient Assisted Living Solutions: The Localization Competition, *IEEE Pervasive Computing*, **12**(4) (2013), 72–79.
- [5] N. Fet, M. Handte, S. Wagner and P. Marron, Enhancements to the LOCOSmotion Person Tracking System, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 72–82.
- [6] A. Braun and T. Dutz, AmbiTrack – Marker-free Indoor Localization and Tracking of Multiple Users in Smart Environments with a Camera-based Approach, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 83–93.
- [7] A. Salem, P. Canalda and F. Spies, A GPS/Wi-Fi/Marker Analysis Based Simultaneous and Hierarchical Multi-Positioning System, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 106–116.
- [8] J. Quintas, A. Cunha, P. Serra, A. Pereira, B. Marques and J. Dias, Indoor Localization and Tracking Using 802.11 Networks and Smartphones, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 117–127.
- [9] G. Pirkl and P. Lukowicz, Indoor Localization Based on Resonant Oscillating Magnetic Fields for AAL Applications, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 128–140.
- [10] G. Pirkl and P. Lukowicz, Robust, low cost indoor positioning using magnetic resonant coupling, in: *Proc. 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 431–440.
- [11] F. Cavallo, M. Aquilano, M. Bonaccorsi, R. Limosani, A. Manzi, M. Carrozza and P. Dario, On the design, development and experimentation of the ASTRO assistive robot integrated in smart environments, in: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4310–4315.
- [12] F. Cavallo, M. Aquilano, M. Bonaccorsi, R. Limosani, A. Manzi, M. Carrozza and P. Dario, Improving Domiciliary Robotic Services by Integrating the ASTRO Robot in an Aml

- Infrastructure, *Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe*, 2014, pp. 267–282.
- [13] A. Moschevikin, A. Galov, A. Soloviev, A. Mikov, A. Volkov and S. Reginya, RealTrac technology overview, *Evaluating AAL Systems Through Competitive Benchmarking*, eds: J.A. Botia et al., **386** (2013), 60–71.
- [14] Evaluating AAL Systems through Competitive Benchmarking. Technical annex of indoor localization and tracking, <http://evaal.aaloa.org/current-competition/technical-annexes-localization2013> (accessed February 2014).
- [15] A. Moschevikin, A. Galov, S. Reginya, A. Volkov and A. Sikora, The Impact of NLOS Components in Time-of-Flight Networks for Indoor Positioning Systems, in: *Proc. IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 2013, pp. 455–460.
- [16] L. Mraz and M. Simek, Visualization and Monitoring Tool for Sensor Devices, *Real-World Wireless Sensor Networks*, 2014, pp. 61–64.
- [17] Developers Handbook – 3rd EvAAL competition. Special theme on Indoor Localization and Tracking, [http://evaal.aaloa.org/images/annex/EvAAL2013/localization/competitors\\_handbook\\_indoor\\_localization\\_2013.pdf](http://evaal.aaloa.org/images/annex/EvAAL2013/localization/competitors_handbook_indoor_localization_2013.pdf) (accessed February 2014).
- [18] Widyawan, M. Klepal and S. Beauregard, A Backtracking Particle Filter for fusing building plans with PDR displacement estimates, in: *Proc. 5th Workshop on Positioning, Navigation and Communication (WPNC)*, 2008, pp. 207–212.
- [19] P. Davidson, J. Collin and J. Takala, Application of particle filters for indoor positioning using floor plans, in: *Proc. Conference on Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2010, pp. 1–4.
- [20] A. Ruiz, F. Granja, J. Prieto Honorato and J. Rosas, Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements, *IEEE Transactions on Instrumentation and Measurement*, **61**(1) (2012), 178–189.
- [21] W. Li, R. Iltis and M. Win, A smartphone localization algorithm using RSSI and inertial sensor measurement fusion, in: *Global Communications Conference (GLOBECOM)*, 2013, pp. 3335–3340.
- [22] M. Atia, M. Korenberg and A. Noureldin, Particle-Filter-Based WiFi-Aided Reduced Inertial Sensors Navigation System for Indoor and GPS-Denied Environments, *International Journal of Navigation and Observation*, <http://dx.doi.org/10.1155/2012/753206> (accessed November 2014).
- [23] A. De Angelis, J. Nilsson, I. Skog, P. Handel and P. Carbone, Indoor positioning by ultrawide band radio aided inertial navigation, *Metrology and Measurement Systems*, **17**(3) (2010), 447–460.
- [24] A. Benini, A. Mancini, A. Marinelli and S. Longhi, A Biased Extended Kalman Filter for Indoor Localization of a Mobile Agent Using Low-Cost IMU and UWB Wireless Sensor Network Robot Control, in: *Proc. 10th IFAC Symposium on Robot Control*, 2012, pp. 735–740.
- [25] S. Pittet, V. Renaudin, B. Merminod and M. Kasser, UWB and MEMS based indoor navigation, *Journal of Navigation*, **61**(3) (2008), 369–384.
- [26] A. Benini, A. Mancini, E. Frontoni, P. Zingaretti and S. Longhi, Adaptive Extended Kalman Filter for Indoor/Outdoor Localization using a 802.15.4a Wireless Network, *ECMR*, 2011, pp. 315–320.
- [27] A. Galov and A. Moschevikin, Bayesian filters for ToF and RSS measurements for indoor positioning of a mobile object, in: *Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013, pp. 310–317.
- [28] A. Mikov, A. Moschevikin, A. Fedorov and A. Sikora, A Localization System Using Inertial Measurement Units from Wireless Commercial Hand-held Devices, in: *Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013, pp. 857–863.
- [29] R. Mahony, T. Hamel and J. Pflimlin, Nonlinear Complementary Filters on the Special Orthogonal Group, *IEEE Transactions on Automatic Control*, **53**(5) (2008), 1203–1218.
- [30] Z. Chen, Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond, <http://www2.ee.kuas.edu.tw/~lwang/WWW/BayesianFilteringFromKalmanFiltersToParticleFiltersAndBeyond.pdf> (accessed November 2014).
- [31] Widyawan, M. Klepal and D. Pesch, A Bayesian Approach for RF-Based Indoor Localisation, in: *Proc. 4th International Symposium on Wireless Communication Systems*, 2007, pp. 133–137.