EVALUATION OF RF-BASED
INDOOR LOCALIZATION SOLUTIONS
FOR THE FUTURE INTERNET

D2.4 Final Version of the EVARILOS Benchmarking Handbook

Author(s): Tom Van Haute (iMinds), Eli De Poorter (iMinds), Ingrid Moerman (iMinds), Filip Lemic (TUB), Vlado Handziski (TUB), Arash Behboodi (TUB), Adam Wolisz (TUB), Niklas Wirström (SICS), Thiemo Voigt (SICS), Pieter Crombez (THC), Gerardo Glorioso (ADV)

Status/Version: Final
Delivery date: February 6th, 2015
Confidentiality: Public
EVARILOS Identifier: EVA-D24-iMinds-V2.1

Keywords:
Indoor localization, benchmarking, metrics, scenarios, methodology, workflow, interference / non-interference, sensitivity, application domains

Abstract:

The EVARILOS Benchmarking Handbook describes the procedures for enabling objective evaluation and comparison of different indoor localization solutions. In this deliverable, we provide a generic methodology for benchmarking of RF-based indoor localization solutions, which is aligned with the upcoming ISO/IEC 18305 standard: “Test and Evaluation of Localization and Tracking Systems”. Moreover, we provide a set of validated and standardized experiment-based benchmarking scenarios focused on the evaluation of RF-based indoor localization solutions in the environments with controlled RF interference. We also provide a set of workflows for accessing some specific properties of RF-based indoor localization solutions, such as sensitivity to interference or influence of the number of anchor nodes on the performance of an indoor localization solution. Finally, we show how the proposed methodology can be used in two specific application domains: healthcare an underground mining.

Table of Contents

List of Abbreviations

AoA :	Angle of Arrival
AoI :	Area of Interest
API :	Application Programming Interface
BLIP :	Bluetooth Local Infotainment Point
CoTS :	Commercial off-the-shelf
CREW :	Cognitive Radio Experimentation World
CSS :	Chirp Spread Spectrum
CSMA/CA :	Carrier Sense Multiple Access met Collision Avoidance
DECT :	Digital Enhanced Cordless Telecommunications
DSSS :	Direct Sequence Spread Spectrum
DTDoA :	Differential Time Differences of Arrival
EVARILOS:	Evaluation of RF-based Indoor Localization Solutions for the Future Internet
EBH :	EVARILOS Benchmarking Handbook
EBS :	EVARILOS Benchmarking Suite
FP7 :	Framework Programme 7
FPGA :	Field-Programmable Gate Array
GSM :	Global System for Mobile Communications
HMM :	Hidden Markov Model
HVAC :	Heating Ventilation Air Conditioning
IEEE :	Institute of Electrical and Electronics Engineers
II :	Interference Impact
IEC :	International Electrotechnical Commission
ISO :	International Organization for Standardization
JTC :	Joint Technical Committee
MIT :	Massachusetts Institute of Technology
mW :	Miliwatt
PDA :	Personal digital assistant
RF :	Radio Frequency
RFID :	Radio Frequency Identifier
RSSI :	Received Signal Strength Indication
RTLS :	Real Time Location System
SC :	Subcommittee
SME :	Small and Medium Enterprises
SoA :	State of the Art
SPIDA :	SICS parasitic interference directional antenna
SUT :	System Under Test
SUT :	Solution Under Test
TDoA :	Time Difference of Arrival
ToA :	Time of Arrival

Continued on next page

ToF :	Time of Flight
TWIST :	TKN Wireless Indoor Sensor Network Testbed
UHF :	Ultra High Frequency
UWB :	Ultra-wideband
WG5 :	Working Group 5
WSN :	Wireless Sensor Network

Chapter 1

Executive Summary

The EVARILOS project addresses one of the major problems of indoor localization: the challenge of reproducing research results in real life and the inability to compare their performance due to evaluation under individual, not comparable and not repeatable conditions. This document presents a benchmarking methodology that remedies these shortcomings. It enables objective experimental validation and fair comparison between different indoor localization solutions in various use-case scenarios and configuration setups. To this end, the document provides a methodology as well as a well-defined set of benchmarking scenarios to evaluate and compare localization solutions. The benchmarks do not focus exclusively on the accuracy of the evaluated localization solution, but also consider e.g. their latency, sensitivity to RF interference and other performance metrics that are relevant in view of the commercial deployment of localization solutions such as complexity, cost, or energy efficiency. The authors would also like to note that the EVARILOS Benchmarking Handbook will be in the future extended in case further practical experience brings some additional hints.

The remainder of the document is organized as follows. Chapter 2 discusses the scope of the EVARILOS Benchmarking Methodology. Next, Chapter 3 focuses on the most important part of the EVARILOS Benchmarking Handbook (EBH): this chapter details how the previous methodology is used to create well-defined benchmarking scenarios. In addition, a list of predefined benchmarking scenarios that can be instantiated in multiple environments is provided, which can be used for objectively comparing different localization solutions. Chapter 4 describes workflows that combine multiple benchmarking scenarios to calculate how the behavior of a localization solution changes under different conditions. Chapter 5 provides different options for assigning application-dependent scores using the output of the previous benchmarking scenarios and workflows. Finally, Chapter 6 concludes the document, while Appendix A describes two relevant application domains, namely healthcare and underground mining. These application domains give us the opportunity to evaluate our handbook in a real life scenario.

Chapter 2

Methodology for Benchmarking of RF-based Indoor Localization

This chapter describes the generic EVARILOS Benchmarking Methodology, which is aligned with the upcoming ISO/IEC 18305 standard: “Test and Evaluation of Localization and Tracking Systems”. The goal of the methodology is to provide suitable steps for enabling fair comparison and evaluation of different localization solutions. In addition, the chapter provides a brief overview of the different approaches for testing and evaluation of indoor localization systems. We argument about the comparative benefits and shortcomings of using integrated system testing vs. component testing and finally we discuss the impact of the clean decoupling between the System Under Test (SUT) and the testing infrastructure.

2.1 Methodology

Contrary to previous approaches, the EVARILOS Benchmarking Methodology does not focus exclusively on the accuracy of the evaluated indoor localization solution, but also considers the latency of the solution, the sensitivity of the solution to RF interference, as well as other performance metrics that are relevant from the point of view of practical deployment of localization solutions such as complexity, cost, energy efficiency, etc.

The EVARILOS Benchmarking Methodology clearly decouples between evaluating individual metrics and the calculation of final scores that are used for ranking different solutions. As illustrated in Figure 2.1, after collecting a set of measurements necessary for the calculation of the individual metrics, the EVARILOS Benchmarking Methodology proposes the usage of weighting factors and thresholding for the calculation of the final ranking scores, reflecting different impacts of the individual metrics for a particular application scenario of interest.

Figure 2.1: The EVARILOS Benchmarking Methodology transforms measurements to scores using multiple evaluation metrics

2.2 Benchmarking Scope

An important decision when designing a benchmarking methodology for the evaluation of indoor localization solutions is the selection between an approach focused on system-level testing, vs. the one that analyses the individual components that typically comprise a localization system.

A system-level testing approach focuses on the localization solution as a whole, without differential treatment of the individual components comprising the system. By concentrating on the performance of the system on the highest functional level, the system testing comes closest to the interests of the end-users of the localization systems, who are mostly interested whether the system as a whole meets their specific requirements. As long as the system performs as expected, the end-users typically are not interested in the internal decomposition of the system.

In contrast to the end-users, the system developers and localization systems researchers might be interested in the structure and the performance of the individual system components. On one hand, a benchmarking methodology focused on the individual components of the system might provide richer insight that can be used to improve the operation of the constituent components. For example, one can concentrate on the evaluation of the ranging component of a localization system, without considering the multilateration-based location estimation component. On the other hand, the results from a strict component-focused testing do not clearly indicate how the system performs as a whole.

Both approaches have their advantages and shortcomings. The EVARILOS Benchmarking Methodology presented in this document focuses on system-level evaluation: complete localization solutions (referred to as Systems Under Test (SUT)) are evaluated using a set of different functional and non-functional metrics. In this way the methodology lays a foundation for comparative evaluation of different localization solutions and their ranking according to a use case specific scoring.

2.3 Level of Coupling with the System Under Test

A second crucial design decision relates to the level of decoupling between the benchmarking procedure and the SUT. Similarly to the differentiation between black-box and gray-box testing in the software engineering, the specification of a benchmarking methodology for indoor localization solutions can be made with or without leveraging the knowledge about the inner workings of the indoor localization SUT.

The knowledge of the internal components of the SUT can be leveraged for designing more efficient benchmarking procedures. For example, if one knows the failure modes of a particular component of a given localization solution, a tailored benchmarking scenario can be designed that evaluates the performance of the SUT under these specific critical operating conditions. Such a customized benchmarking approach can provide better insight in the behavior of the solution, and can be used to optimize the testing process by focusing on the parameter space that is relevant to a specific SUT.

The main drawback of a SUT-specific benchmarking is the loss of generality, since the test scenarios have to be customized to each SUT. By following a black-box testing approach, that ignores the internal mechanisms of a system or component and focuses solely on the outputs generated in response to selected inputs [1], one can test different SUTs against common functional requirements. The EVARILOS Benchmarking Methodology follows this approach and uses generic testing scenarios that can be applied to different SUTs. The testing scenarios are carefully orchestrated to guarantee fair comparison between heterogeneous SUTs by exploring different failure modes of various RF-based indoor localization solutions.

The benchmarks described in this deliverable are implemented in the EVRILOS benchmarking suite deliverable (D2.5) [8]. This benchmarking suite provides reference benchmark scenarios, reference code for calculating metrics and scores and publicly makes available all results from EVARILOS benchmarks. Full implementation details are available in deliverable D2.5.

2.4 Benchmarking Terminology

This section describes the terminology used throughout the remainder of this document. Definitions built further upon existing work about benchmarking methodologies for wireless networks, mainly from the FP7 CREW project [2] and the ISO Standard [4]. The list can be found in Table 2.1 and a graphical overview can be found in Figure 2.2 [5].

Table 2.1: Benchmarking terminology

TERMS	EXPLANATION
Benchmark	Well-defined procedure for executing an experiment with SUT, collecting data and calculating metrics.
Configuration	Set of parameters describing conditions of the experiments, e.g interference conditions, evaluation points, people, night/day.
Environment	The external context in which a benchmarking experiment is performed, including also factors that cannot be actively influenced. The environment is assumed to be stable up to a certain, uncontrollable randomness.
Benchmarking scenario	Combination of benchmark, environment and configuration.
Experiment	Execution of a benchmarking scenario.
Ground truth	Refers to the conceptual idea of an ideal solution that gives exact location estimates without any communication overhead or delay. The performance of a localization solution is often compared with the ground truth.
Metric	Indicator of performance of the SUT.
Score	A function defined over a set of metric. A typical function might be a weighted linear combination. Scores are useful for ranking the performance of different SUTs for specific use-cases.
Reference scenario	Benchmarking scenario that is used as a reference, i.e. benchmarking results obtained by executing this benchmarking scenario are used for comparison with benchmarking results of other benchmarking scenarios executions.
Repeatability	Implies that different executions of the same benchmarking scenario obtain comparable evaluation metrics under well determined conditions. This comparability is however not strict in wireless benchmarking due to a certain level of indeterminism in the wireless environment. For repeatability to apply, acceptable error margins should be formally defined.
Reproducibility	Extension on repeatability, where different executions of the same benchmarking scenario in different environments should result in comparable benchmarking results. The same error margins on equality apply as in repeatability.

Continued on next page

Table2.1 – continued from previous page

TERMS	EXPLANATION
System Under Test	Refers to a system whose performance is to be evaluated and possibly compared to the performance of other SUTs.
Testbed	Facility that enables a long term and sustainable benchmarking. Note that a benchmark does not necessarily need a testbed for its execution.
Use-case	Or application domain, drives the selection of scenarios and final scores.
Workflow	Repeated series of experiments aiming at answering specific questions. Typical questions are: achieving statistical significance of the results, investigating the sensitivity of the SUT on some conditions.
Benchmarking results	Set of metrics that are the results of the execution of a benchmarking scenario.

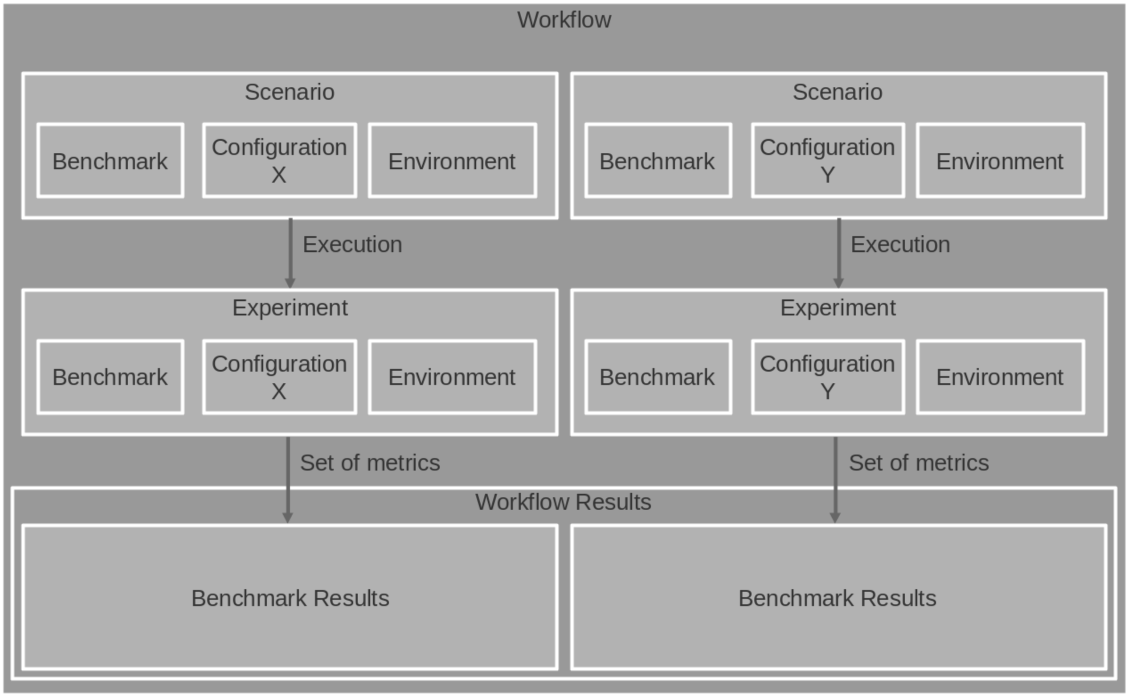


Figure 2.2: Graphical overview of the methodology for obtaining the benchmarking results by executing a benchmarking scenario

2.5 Relation with the ISO/IEC 18305

In parallel to the EVARILOS project, ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) have established a technical committee, ISO/IEC JTC 1, to jointly work on drafts with the aim of standardizing “Test and evaluation of localization and tracking systems”. These draft are currently being prepared by the Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 31, Automatic identification and data capture techniques, Working Group 5, Real time locating systems [3]. The committee is referred to as ISO/IEC JTC1/SC31/WG5.

The committee is currently working on two types of standards. A first class of draft standards focuses on determining the suitability of single-technology equipment for localization purposes. These documents describe international standard compliance test methods for determining localization performance characteristics through objectively evaluation of link metrics such as packet error rate, ranging distance, latency, influence of different orientations, ranging success rate, etc. As such, the main purpose of these drafts and work items is to establish whether a component is suitable for integration in an overall localization solution. Examples include e.g. ISO/IEC FDIS 24770-62 (Evaluating performance characteristics of High rate pulse repetition frequency Ultra Wide Band (UWB) air interface equipment), ISO/IEC FDIS 24770-62 (Evaluating performance characteristics of Chirp Spread Spectrum (CSS) Real Time Locating System (RTLS)) and ISO/IEC 24769-22 (Test methods for air interface communication at 2.4 GHz). In contrast to the EVARILOS project, no full-systems are evaluated and the evaluation metrics do not focus on large-scale deployments in different conditions.

The ISO/IEC 18305 draft [4] is more in line with the EVARILOS goals. The draft is at the time of writing not yet publicly available, but includes a taxonomy of localization solutions and describes a wide range of evaluation scenarios and performance metrics for these solutions. In contrast to the EVARILOS project, which focuses mainly on RF-based localization solutions, the draft also considers indoor localization solutions that use a wide range of other sensors such as accelerometers. Several of the evaluation scenarios include difficulties designed to stress test sensor based tracking systems, for example by including scenarios that include crawling through hallways. The terminology and metrics from the ISO/IEC 18305 and the EVARILOS handbook are currently, as much as possible considering their different scope, aligned. In addition, EVARILOS goes beyond this work by including non-functional metrics (such as deployment metrics), by providing score calculations, by automating the benchmarking process and by giving public access to the results and data to a wide community.

Chapter 3

EVARILOS Benchmarks

Whereas the previous chapter described a very generic methodology for evaluating localization solutions, the goal of the EVARILOS Benchmarking Handbook (EBH) is to describe a list of well-defined benchmarks. A benchmark can be considered as an instantiation of the EVARILOS Benchmarking Methodology: it is a very specific procedure for obtaining predefined performance metrics. When instantiated in a specific environment using pre-defined configuration settings, a benchmark is referred to as a benchmark scenario. This chapter will first describe the components of a benchmarking scenario, after which a list of reference benchmarks will be given.

3.1 Components of a Benchmarking Scenario

This section describes the general components that are required to define a benchmarking scenario. A benchmarking scenario consists of a combination of (i) environment specifications, (ii) a configuration (e.g. setup descriptions) and (iii) a benchmark (describing the evaluation procedure and the obtained metrics). The environmental specifications define both structural properties (e.g. room sizes, wall types) and RF interference properties, i.e. what types of external uncontrollable interference sources are present and to what extent. The configuration description defines the required properties of a localization solution itself. This includes, e.g. the amount of anchors to be deployed and locations and time for taking the measurements for training of the parametrization of an indoor localization solution. The metrics specification defines the evaluation procedure, such as which metrics are relevant and will be obtained in a particular benchmark.

All components are considered in a black-box fashion: the benchmark description can be seen as a black box, which takes as input a localization approach, and outputs one or more numerical benchmarking metrics. As such, the internal properties of the localization benchmark are not evaluated, only its relevance for different application domains.

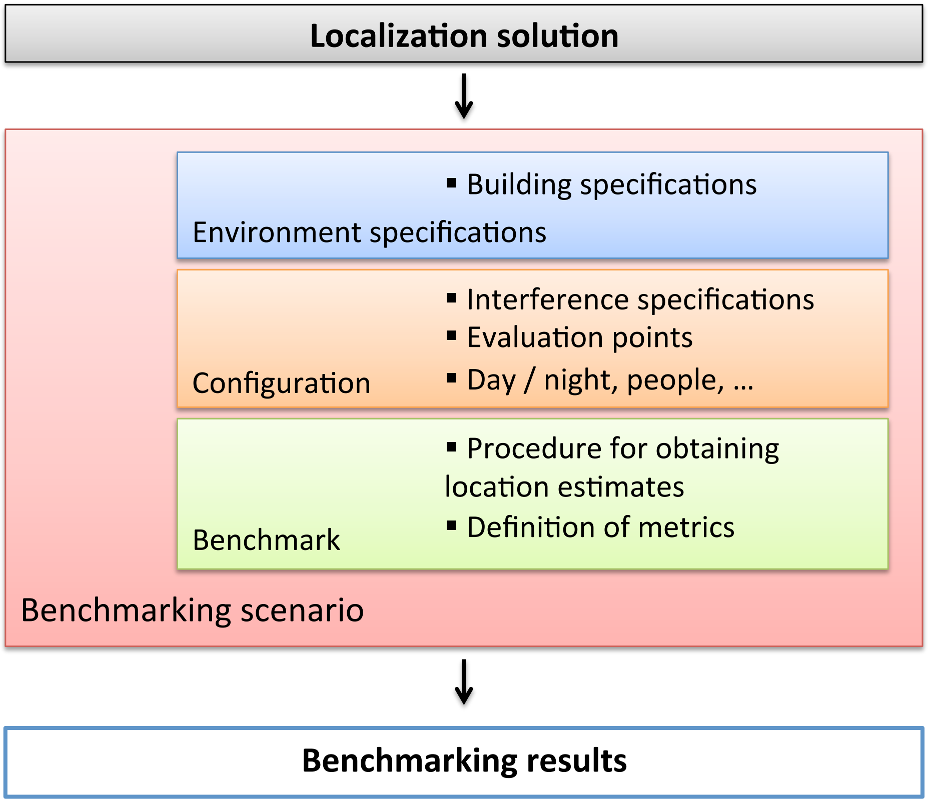


Figure 3.1: Components of a benchmarking scenario

Execution an experiment that corresponds to a benchmarking scenario is done under the assumption that the environment is stable: the tester tries to maintain the same conditions of the environment during the test. For example, if a certain experiment scenario is without interference, this condition has to be maintained before, during and also after the experiment. If interference is detected, the experiment is considered invalid and has to be re-executed.

To illustrate the different components of a benchmarking scenario, an example evaluation is shown below. Consider a company that wants to test the suitability of a localization solution under interference for localizing persons in an exhibition. The **necessary input** of the company is a localization solution under test. This includes a description (from the provider of the SUT) about how many anchor points are needed and where and how they have to be set up. Preferably, also detailed descriptions about the hardware, software and algorithm are available. The benchmarking scenario consists of the following components:

- The company would like to test their solution for an exhibition, so the **environment** has the following specifications:
  - The **building specification** is an open space.
  - The **interference specifications** in this situation are comparable with the office interference.
  - The **size** of the environment is big.
- The **interference** should be representative to the one found in typical exhibition environments.
- Finally, the **evaluation procedure** needs to take into account typical behavior of exhibition visitors.

- The next step is the specification of the **evaluation points**. The evaluation point specification describes at which locations the SUT is tested and describes the evaluation procedure at each evaluation point. Two options are possible, on the one hand, a good scenario uses localizations that are representative for typical usage in the application domain. On the other hand, it is also possible that the evaluation points are chosen randomly for a general evaluation.
- Finally, the **evaluation metric** describes in detail which key performance indicators are evaluated and how they are calculated. In this example, the accuracy and delay metric are selected.

The output of the above benchmarking scenario are the evaluation metrics. As described in the previous chapter, these evaluation metrics can optionally be transformed into scores as described in the **evaluation criteria**. Score calculation is discussed in Chapter 5. In the next sections, each component of a benchmarking scenario will be described in more detail.

3.1.1 Indoor Localization Solution

The input for the localization benchmarking is a complete solution. To this end, the installation guidelines / recommendations from the provider of the solution (number of anchor nodes etc.) are required. Further, the EVARILOS Benchmarking Suite provides an API that can be used to integrate the SUT for the evaluation according to the EVARILOS Benchmarking Methodology, with more details given in the deliverable D2.5. EVARILOS Benchmarking Suite.

3.1.2 Environment Specifications

Environments consist of building specifications (adapted for different sizes) and interference specifications.

3.1.2.1 Building Specifications

The most typical characteristic of a building is the type of wall. In the EVARILOS Benchmarking Handbook, three different types of walls are distinguished: open space, no walls (Table 3.1), (ply)wooden walls (Table 3.2) and brick walls (Table 3.3). For each type of wall, a corresponding room size must be selected (small, medium or big). Since the performance of an indoor localization solution is often strongly related to the type of environment, all benchmarking outputs must always be given together with a description of the building specifications. For a fair comparison, the EVARILOS Benchmarking Handbook describes in detail a number of predetermined reference building types.

In some situations, a combination of different wall types is necessary. In that case, a certain percentage of multiple wall types can be combined. Depending on the evaluation criteria, the weighted average or the minimum value will be used to determine the final score.

Table 3.1: Open space: specifications

	Small	**Medium**	**Big**
PARAMETER	VALUE		
Number of rooms	1	1	1
Minimum area (m^2)	20	100	2 000
Maximum area (m^2)	100	2 000	10 000
Example	Meeting room	Conference room	Exhibition hall

Table 3.2: Wooden walls: specifications

	Small	**Medium**	**Big**
PARAMETER	VALUE		
Minimum number of rooms	2	20	50
Minimum area (m^2)	5	10	100
Maximum area (m^2)	10	100	1 000
Example	Small office	Big office	Museum exhibition

Table 3.3: Brick walls: specifications

	Small	**Medium**	**Big**
PARAMETER	VALUE		
Minimum number of rooms	10	20	100
Minimum area (m^2)	50	100	400
Maximum area (m^2)	100	400	5 000
Example	House	Villa / small office	Hospital

3.1.2.2 Interference Specifications

The interference specifications describe the presence of RF signals in an environment. In Table 3.4 four tables with interference specifications are given. The interference will be defined by the type and parameters of interference source, network parameters and traffic parameters of the interference. When interference is present, these parameters should be clearly specified.

Table 3.4: Interference: specifications

TYPES OF INTERFERENCE SOURCE
Microwave
WiFi
DECT
Bluetooth
3G
Zigbee

PARAMETERS OF THE INTERFERENCE SOURCE
Number of sources
Power
Waveform
Specific pattern
Start & stop time
Traffic model

TRAFFIC PARAMETERS OF THE INTERFERENCE
Packet size
Inter packet gap
Bitrate
File size
Start & stop time
Traffic model

NETWORK PARAMETERS
Network size
Node density
Node mobility
Node failures

3.1.3 Evaluation Points Specification

In this section, different selection sampling techniques are explained in order to define the EVARILOS Benchmarking Scenarios. These sampling techniques are evaluated and discussed in the following subsections. As illustrative test case, the 3^{rd} floor of the w-iLab.t I testbed in Ghent is used. There are several ways to chose a set of the evaluation points. Generally, they can be divided into two groups:

- Evaluation points are chosen in a random fashion.
- Evaluation points are based on a specific use case.

3.1.3.1 Defining Evaluation Points Randomly

Random Sampling

The simplest objective technique to select measurement points is by selecting them randomly. An example is given in Figure 3.2.

Figure 3.2: Random sampling: a “good” example

The advantage is the simplicity of this algorithm. The pseudo code of this algorithm is shown in Listing 3.1.

Listing 3.1: Random algorithm: pseudo code

```
xMax = width of the map
yMax = height of the map
data = list of points

foreach(# measurement points)
{
        x = random 0 → xMax
        y = random 0 → yMax

        add new point (x,y) to data
}
```

However, this algorithm has multiple drawbacks:

- The measurement points are not always equally distributed over the entire area. The bigger the area, the higher the probability that certain parts of the area will not be used for testing.
- Some generated measurement points are feasible not reachable (e.g. at the center of a wall).

The drawbacks are shown in Figure 3.3. The left hand side and center part of the floor are not equally distributed in comparison with the right hand side. Besides, there are also three points generated in the center of a wall, which is not usable for a scenario.

Figure 3.3: Random sampling: a “bad” example

Grid Sampling

An optimization in order to solve the second drawback of the previous algorithm, is to use a grid (Figure 3.4). In this case, the advantage is that the possibilities are limited. Only the intersections can act as a possible measurement points. Due to that, the grid can be defined in a way that the intersections never overlap with a wall or impossible place for measurements.

Figure 3.4: Grid sampling

In this algorithm, a set of measurement points is given (the intersections of the grid) and the indexes are selected by randomness (Listing 3.2).

Listing 3.2: Grid algorithm: pseudo code

```
xMax = number of columns of the matrix
yMax = number of rows of the matrix
gridData = matrix containing the coordinates of the intersections
data = list of measurement points

foreach(# measurement points)
{
        x = random 0 → xMax
        y = random 0 → yMax

        while((x,y) already used)
        {
                x = random 0 → xMax
                y = random 0 → yMax
        }
```

```
        get point A = gridData(x,y)
        add point A to data
}
```

Still, due to the randomness the results do not guarantee a desired distribution (an example is given in Figure 3.5).

Figure 3.5: Grid sampling with randomness

Latin Hypercube
The Latin hypercube sampling technique uses the principle of a Latin square. The most remarkable characteristic of a Latin square is that each element only appears once in each column or row. But in this case, we do not have a square but a rectangle so the maximum of measurement points is up to five.

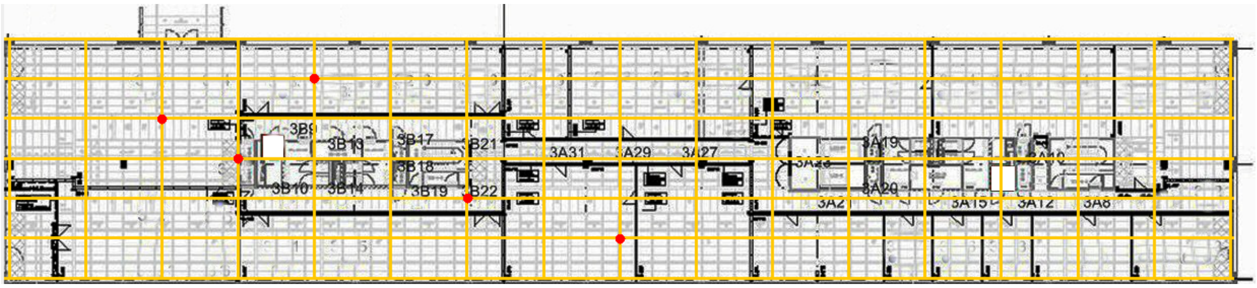


Figure 3.6: Latin hypercube

Listing 3.3: Latin hypercube algorithm: pseudo code

```
xMax = number of columns of the matrix
yMax = number of rows of the matrix
gridData = matrix containing the coordinates of the intersections
data = list of measurement points

if(# measurement points > # rows)
        # measurement points = # rows

foreach(number of measurement points)
{
        x = random 0 → xMax
        y = random 0 → yMax

        while((x column or y row already used)
        {
```

```
                x = random 0 → xMax
                y = random 0 → yMax
        }

        get point A = gridData(x,y)
        add point A to data
}
```

Multi-sector Latin Hypercube Sampling
The final optimization of the sampling algorithm is the subdivision of the total space in equal subspaces. The long rectangle can be divided in three squares. Next, the Latin hypercube sampling technique is applied on the three squares separately. The result of this algorithm can be found in Figure 3.7. The yellow lines separate the squares. In this situation, 15 different measurement points can be generated.

Figure 3.7: Advanced latin hypercube sampling

Optimal Evaluation Points Sampling Approach

The above mentioned approaches each have advantages and disadvantages. Random sampling is not advised: the points are not equally distributed and can possibly include unfeasible locations. In general, uniform sampling approaches are preferred, since they better represent the overall environment in which the solution is evaluated. The advantages of the orthogonal sampling technique are the following:

- The algorithm permits random sampling.
- On resulting evaluation points are feasible.
- The total number of possible evaluation points is limited and known.
- The result is a well-spread distribution of the evaluation points.

When using the orthogonal sampling technique (and by using a well defined grid structure), it is possible to limit the number of evaluation points to a subset of locations that are all reachable. These points can be preprogrammed in the mobile devices, or can be marked to facilitate manual testing. As such, the orthogonal sampling technique is a good trade-off between simplicity and uniformness to create a fixed set of evaluation points. For multiple evaluations in the same environment, orthogonal sampling can create multiple valid sets each time with different randomly selected measurement points.

Figure 3.8: Advanced latin hypercube sampling: final result

3.1.3.2 Definition of Evaluation Points based on a Specific Use-Case

In this section, an example is given that the evaluation points are based on a certain use-case: in Figure 3.9 a traveling scenario is proposed. This scenario describes one of the most common cases in an office environment. A person will be traced during his daily tasks, let us call him “Jan” for simplicity.

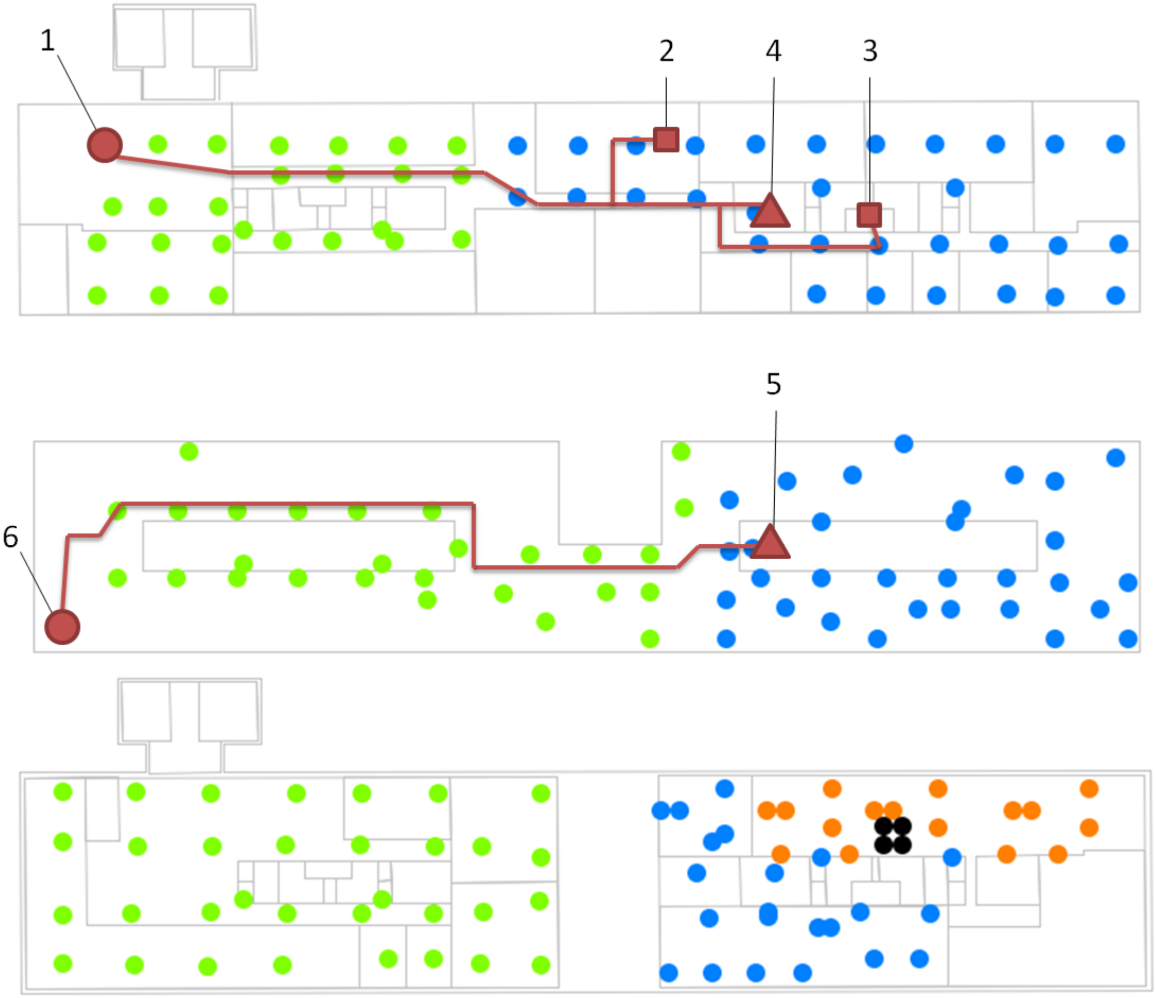


Figure 3.9: The w-iLab.t I wireless testbed: map & scenario description

The measurements start when Jan takes some papers from the printer. The printer is located at point 1. Then he walks back to his office and works for a couple of minutes (point 2). Both the printer and his office are located at the third floor. Thereafter, Jan has a meeting at point 6 on the map. The meeting room is located at the second floor, but first, Jan wants to refill his cup of coffee in the kitchen (point 3). He is using the elevator at point 4 and 5.

The average speed of Jan (and the localized node) is about 3 km/h. He will also walk close to the walls sometimes (mainly in the corridors) because other people are crossing. Furthermore, this action will be repeated 10 times, in both directions.

The floor plan of the second floor is not detailed, but here are corridors and many different rooms as well. The corridor is in the center of the building surrounded by all small office and meeting rooms.

Table 3.5: Detailed description of the location of the measurement points

POINT	LOCATION DESCRIPTION
1	Point 1 is located on the third floor close to the printer. The exact location is in front of office 3.18. This point is right under the sensor node and one meter removed from the wall.
2	The second point is also located at the third floor, in office 3.16, close to the desk in the right corner. The point is set two meters away from both walls.
3	Point 3 is pinned at the eastern side of the building, in the center of the kitchen. The kitchen is a small room where no sensor node is available and there is a microwave.
4	This is the last point at the third floor, it is in front of the elevators (about one meter away). Because there are two elevators, the center of both is chosen.
5	The coordinates of point 5 are exactly the same as point 4, only point 5 is located on the second floor.
6	The trace ends in meeting room 2.30 (called Bell). The precise location of point 6 is in the corner by the window, also removed two meters of both walls.
1 → 7	Every measuring point is situated about 1.5 meters above the ground.

3.1.4 Evaluation Metrics

Every benchmarking scenario contains a number of metrics that will be used for evaluating an indoor localization solution. This section provides detailed descriptions of the individual metrics comprising the EVARILOS Benchmarking Methodology. A metric is a measure of the performance of the localization SUT. They are needed to calculate a final score for an indoor localization solution: once the metrics have been calculated, they can be combined in a final use case score using specific weighting factors that reflect the importance of the individual metrics in the particular use-case of interest.

For each individual metric, a definition is given, together with instructions for collecting the necessary underlying measurements and a mathematical formula (where applicable) that should be used for processing those measurements in order to calculate the metric value. The metrics are organized in two generic categories: performance metrics and deployment metrics. The first and largest category is comprised by several metrics that try to capture different performance aspects of the SUT, such as its accuracy, latency, scalability, etc. In contrast, the deployment metrics focus on non-performance related attributes such as the underlying technology, licensing modalities, efforts and costs needed for installation and configuration of the SUT.

3.1.4.1 Metrics Summary

On the next page, in Table 3.6 an overview of all the metrics is given. A graphical overview is shown in Figure 3.10 describing the dependencies and coherence of the different metrics.

Table 3.6: Overview of all metrics

TYPE	NAME	SUBMETRICS	SHORT DESCRIPTION	METRIC UNIT
Performance	Point accuracy		Point accuracy implies the Euclidean error distance between a reference and a measured point.	distance (cm)
	Room accuracy		The coordinates of the measured point are validated by checking the correctness of the room.	percent (%)
	Latency		The time required to produce a location on request.	time (ms)
	Energy efficiency	• Infrastructure nodes • Localized nodes	The energy consumption of the nodes.	• milliwatt (mW) • milliwatt (mW)
	Interference sensitivity		The change of the performance metrics under different wireless interference conditions.	percent (%)
	Environmental sensitivity		The change of the performance metrics in different physical test environments.	percent (%)
	Sensitivity to mobility		This metric measures the performance for different speeds of the localized node.	percent (%)
	Scalability		This metric measures the performance for different amounts of localized nodes.	percent (%)
	Repeatability		The stability of the solution	percent (%)
Deployment	Set-up overhead	• Physical installation • Configuration • Information needed in advance • replacement time	The set-up overhead includes different factors that influence the initial delay of the start-up.	• time (s) • none • none • time (s)
	Technology		Describes the technology and hardware that is used.	none
	Financial cost	• Installation and fixed nodes • Localized nodes • Maintenance cost	The hardware and maintenance costs of the complete system.	euro (€)

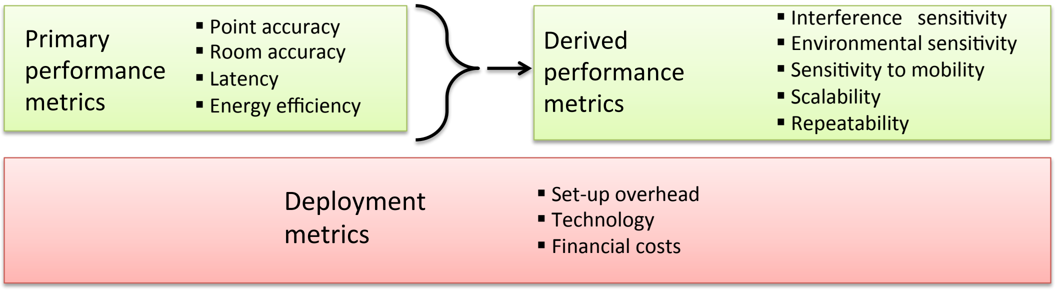


Figure 3.10: Graphical overview of the metrics

3.1.4.2 Performance Metrics

This section describes the mathematical formulas that are used to convert measurements into performance metrics.

Point Accuracy

Description The point accuracy is one of the most important metrics in the EVARILOS Benchmarking Handbook. There are two different accuracy metrics: point and room accuracy. With point accuracy, the actual Euclidean error distance between a reference point and a measured point is calculated. The coordinates of the points have two (x, y) or three (x, y, z) dimensions. The summary of this metric can be found in Table 3.7.

Table 3.7: Point accuracy: summary

NAME	Point accuracy
TYPE	Performance
SUBMETRICS	-
SHORT DESCRIPTION	Point accuracy implies the Euclidean error distance between a reference and a measured point.

Measurement Method To measure the distance between two points, the Euclidean distance equation is used. Suppose the reference point has coordinates (x_1, y_1, z_1) and the measured point (x_2, y_2, z_2), then the error distance d can be found by using Equation 3.1 for 2D and Equation 3.2 for 3D.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.1}$$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{3.2}$$

Once the distances of the multiple tests are calculated the mean, standard deviation, minimum and maximum values can be calculated using the following equations:

$$\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{3.3a}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \overline{d})^2} \tag{3.3b}$$

$$d_{min} = \min(d_1, d_2, ..., d_n) \tag{3.3c}$$

$$d_{max} = \max(d_1, d_2, ..., d_n) \tag{3.3d}$$

Measurement Unit Due to the fact `distance` is used, the unit will be in `centimeter`.

Room Accuracy

Description The room accuracy metric is a variant of the previous one. The coordinates of the measured point will not be compared with the reference point. The coordinates are validated by checking the correctness of the corresponding room. A distinction is made between the different floor levels. In this metric the coordinates also have three dimensions (x, y, z). The summary of this metric can be found in Table 3.8.

Table 3.8: Room accuracy: summary

NAME	Room accuracy
TYPE	Performance
SUBMETRICS	
SHORT DESCRIPTION	The coordinates of the measured point are validated by checking the correctness of the room.

Measurement Method It is not possible to use an exact measurement tool. The measured points will be mapped on a map where the different rooms are visible. Then somebody will check if the room that contains the measured point corresponds with the actual room (at each floor).

To visualize the results of the room accuracy, a room confusion matrix will be used. Each column of the matrix represents the instances in a predicted room, while each row represents the instances in an actual room. An example of a confusion matrix is given in Table 3.9 on the assumption that each room is located next to each other and each room is tested 10 times. A basic floor plan of these rooms is illustrated in Figure 3.11.

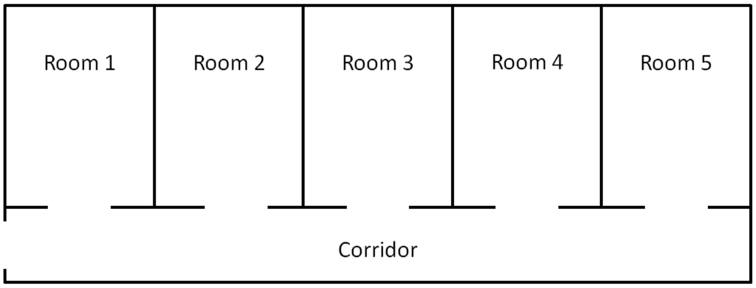


Figure 3.11: Example of five different rooms

Table 3.9: A confusion matrix: example

		Predicted room				
		Room 1	Room 2	Room 3	Room 4	Room 5
Actual room	Room 1	**7**	2	1	0	0
	Room 2	1	**8**	1	0	0
	Room 3	1	2	**6**	0	1
	Room 4	0	1	0	**9**	0
	Room 5	0	0	2	1	**7**

In this table, it becomes clear that the number of correct rooms is in bold (the predicted room corresponds to the actual room). The other numbers are the amount of incorrectly predicted rooms.

With these numbers, a simple success rate can be calculated by dividing the number of correct rooms by the total number of rooms available. This becomes clear in Equation 3.4. Even more sophisticated success rate equations can be used where the geographical position of the rooms can be taken into account.

$$sr = \frac{\textit{number of correct rooms}}{\textit{total number of rooms}} \tag{3.4}$$

Measurement Unit Because the `success rate` is a relative number, this value is expressed in `percent` (%).

Latency

Description This metric defines the latency of the system. This metric is especially relevant for solutions that are deployed in time-critical use cases, such as alarm triggers. Typically, a trade-off exists between this metric on one hand and the energy efficiency and accuracy on the other hand.

Table 3.10: Set-up overhead: summary

NAME	Latency
TYPE	Performance
SUBMETRICS	
SHORT DESCRIPTION	The amount of time needed to locate a certain node when an alarm triggers.

Measurement method This metric expresses the time interval between a request for a location estimate and the reception of the location estimate.

Measurement Unit This metric is defined by an `amount of time`, expressed in `milliseconds` (s).

Relevant statistics for the latency are the mean, standard deviation, minimum and maximum values. These can calculated using the following equations:

$$\overline{r} = \frac{1}{n}\sum_{i=1}^{n} r_i \tag{3.5a}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \overline{r})^2} \tag{3.5b}$$

$$r_{min} = \min(r_1, r_2, ..., r_n) \tag{3.5c}$$

$$r_{max} = \max(r_1, r_2, ..., r_n) \tag{3.5d}$$

Energy Efficiency

Table 3.11: Energy efficiency: summary

NAME	Energy efficiency
TYPE	Performance
SUBMETRICS	• Infrastructure nodes • Localized nodes
SHORT DESCRIPTION	The energy used by the entire SUT.

General

Measuring energy efficiency is a difficult task. At best, it is a single device whereby the energy efficiency can be measured easily. In case of smart phones and laptops, the energy measurement is biased by other applications, OS, etc. Therefore, calculating the marginal cost (in energy) is the best approach. This implies that only the energy used for localization as an increment to the energy used by a set of applications, is charged.

Infrastructure Nodes

Description Energy efficiency is a relevant metric since low energy efficiency allows easier deployment of infrastructure nodes by using alternative energy sources. For those reasons the energy must be used as efficient as possible. The energy efficiency is measured over all infrastructure nodes and then averaged.

Measurement Unit The result of the measurement is `power` and is expressed in `milliwatt` (mW). The unit, defined as one joule per second, measures the rate of energy conversion or transfer. $W = \frac{J}{s}$. Statistics are calculated based on the energy efficiency of all infrastructure nodes. These measurements cover the time intervals when the system is up and running.

$$\overline{W} = \frac{1}{n}\sum_{i=1}^{n} W_i \tag{3.6a}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(W_i - \overline{W})^2} \tag{3.6b}$$

$$W_{min} = \min(W_1, W_2, ..., W_n) \tag{3.6c}$$

$$W_{max} = \max(W_1, W_2, ..., W_n) \tag{3.6d}$$

Localized Nodes

Description Energy efficiency is especially important for the localized nodes since these typically use batteries.

Measurement Method The energy efficiency can be measured using special hardware equipment, or by using theoretical calculations.

Measurement Unit The result of the measurement hardware is `power` and is expressed in `milliwatt` (mW). The unit, defined as one joule per second, measures the rate of energy conversion or transfer. $W = \frac{J}{s}$.

3.1.4.3 Derived Performance Metrics

Derived performance metrics express the sensitivity of primary performance metrics to different conditions. Whereas the primary performance metrics can be calculated using a single benchmark, calculating derived performance metrics requires the use of workflows that combine the output from multiple benchmarks. As such, the calculation of derived metrics consists of at least two phases. During the first phase a metric is calculated in well-described conditions. During the later phases, the mean, standard deviation, minimum and maximum value of metrics are calculated in changing conditions and compared with the original metric of Section 3.1.4.2. Since derived metrics utilize multiple benchmarks, details on how to calculate the derived performance metrics is given both here and in Chapter 4, while their definition is given in the following text below.

Table 3.12: Derived performance metrics: summary

INTERFERENCE SENSITIVITY	The interference sensitivity is evaluated by identifying the difference in primary metrics to different benchmarks with a different amount of interference.
ENVIRONMENTAL SENSITIVITY	The environmental sensitivity is evaluated by identifying the difference in primary metrics to different test environments.
SENSITIVITY TO MOBILITY	The sensitivity to mobility is evaluated by identifying the difference in primary metrics to different speeds of the localized node.
SCALABILITY	The scalability is evaluated by identifying the difference in primary metrics to the presence of different numbers of localized nodes.
REPEATABILITY	The repeatability is evaluated by identifying the difference in primary metrics to multiple repetitions of a same benchmarking scenario.

Table 3.13: Interference sensitivity: summary

NAME	Interference sensitivity
TYPE	Derived performance
SUBMETRICS	
SHORT DESCRIPTION	The interference sensitivity is evaluated by identifying the difference in primary metrics due to different benchmarks with a different amount of interference.

Description The accuracy metric will be used and compared under different circumstances. Different kinds and amounts of interference will be used. Specific types like microwaves but also synthetic interference will be used. How this interference is applied in the environment will be explained in each scenario.

Measurement Method Like it was already mentioned in the introduction, the method consists of two phases.

1. A study of the accuracy in function of the interference can be made using the Equations 3.7a, 3.7b, 3.7c and

3.7d. These are respectively the mean value, the standard deviation, the minimum and maximum value.

$$\overline{x}_{derived} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.7a}$$

$$\sigma_{derived} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.7b}$$

$$x_{min\ derived} = \min(x_1, x_2, ..., x_n) \tag{3.7c}$$

$$x_{max\ derived} = \max(x_1, x_2, ..., x_n) \tag{3.7d}$$

2. The final step is to compare $\overline{x}_{derived}$, $\sigma_{derived}$, $x_{min\ derived}$ and $x_{max\ derived}$ with the original parameters and calculate the difference.

$$\overline{x}_{final} = \frac{\overline{x}_{derived} - \overline{x}}{\overline{x}} \tag{3.8a}$$

$$\sigma_{final} = \frac{\sigma_{derived} - \sigma}{\sigma} \tag{3.8b}$$

$$x_{min\ final} = \frac{x_{min\ derived} - x_{min}}{x_{min}} \tag{3.8c}$$

$$x_{max\ final} = \frac{x_{max\ derived} - x_{max}}{x_{max}} \tag{3.8d}$$

Measurement Unit Because the `interference sensitivity` is a relative number, this value is expressed in `percent` (%).

3.1.4.4 Environmental Sensitivity

Table 3.14: Environmental sensitivity: summary

NAME	Environmental sensitivity
TYPE	Derived performance
SUBMETRICS	
SHORT DESCRIPTION	The environmental sensitivity is evaluated by identifying the difference in primary metrics due to different test environments.

Description The environmental sensitivity defines to which extent a solution is stable for operating in different environments. The variation of the performance metrics needs to be limited.

Measurement Method Performance metrics are required to evaluate this metric. If these values are available for different test environments, then a comparison can be made. Also here, two phases are distinguished:

1. In this way multiple statistical numbers can be calculated (see Equations 3.9a, 3.9b, 3.9c, 3.9d).

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.9a}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.9b}$$

$$x_{min} = \min(x_1, x_2, ..., x_n) \tag{3.9c}$$

$$x_{max} = \max(x_1, x_2, ..., x_n) \tag{3.9d}$$

2. Also here, the final step is to compare the values by subtraction:

$$\overline{x}_{final} = \frac{\overline{x}_{testbed\ 1} - \overline{x}_{testbed\ 2}}{\overline{x}_{testbed\ 1}} \tag{3.10a}$$

$$\sigma_{final} = \frac{\sigma_{testbed\ 1} - \sigma_{testbed\ 2}}{\sigma_{testbed\ 1}} \tag{3.10b}$$

$$x_{min\ final} = \frac{x_{min\ testbed\ 1} - x_{min\ testbed\ 2}}{x_{min\ testbed\ 1}} \tag{3.10c}$$

$$x_{max\ final} = \frac{x_{max\ testbed\ 1} - x_{max\ testbed\ 2}}{x_{max\ testbed\ 1}} \tag{3.10d}$$

Measurement Unit Because the `environmental sensitivity` is a relative number, this value is expressed in `percent` (%).

3.1.4.5 Sensitivity to Mobility

Table 3.15: Sensitivity to mobility: summary

NAME	Sensitivity to mobility
TYPE	Derived performance
SUBMETRICS	
SHORT DESCRIPTION	The Sensitivity to mobility is evaluated by identifying the difference in primary metrics due to different speeds of the localized node.

Description The sensitivity to mobility metric defines the variation of the performance metrics when the speed of the localized node increases / decreases. In the “normal situation” the metrics will be evaluated when the localized node has a speed of 0 km/h. In this derived metric, the node will have a speed of 5 km/h.

Measurement method Performance metrics will be calculated/measured in two scenarios with the same configuration. The only difference between the two scenarios, is the speed of the localized node. A comparison between the same metrics of both scenarios will be made and evaluated.

1. In this way multiple statistical numbers can be calculated.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \tag{3.11a}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3.11b}$$

$$x_{min} = \min(x_1, x_2, ..., x_n) \tag{3.11c}$$

$$x_{max} = \max(x_1, x_2, ..., x_n) \tag{3.11d}$$

2. Also here, the final step is to compare the values by subtraction:

$$\overline{x}_{final} = \frac{\overline{x}_{0\ km/h} - \overline{x}_{5\ km/h}}{\overline{x}_{0\ km/h}} \tag{3.12a}$$

$$\sigma_{final} = \frac{\sigma_{0\ km/h} - \sigma_{5\ km/h}}{\sigma_{0\ km/h}} \tag{3.12b}$$

$$x_{min\ final} = \frac{x_{min\ 0\ km/h} - x_{min\ 5\ km/h}}{x_{min\ 0\ km/h}} \tag{3.12c}$$

$$x_{max\ final} = \frac{x_{max\ 0\ km/h} - x_{max\ 5\ km/h}}{x_{max\ 0\ km/h}} \tag{3.12d}$$

Measurement unit Because it is a relative number, this value is expressed in `percent` (%).

3.1.4.6 Scalability

Table 3.16: Scalability: summary

NAME	Scalability
TYPE	Derived performance
SUBMETRICS	
SHORT DESCRIPTION	The scalability is evaluated by identifying the difference in primary metrics to the presence of different numbers of localized nodes.

Description This metric is comparable with the previous one. The only difference here is the amount of localized nodes instead of the speed that varies. Instead of one single node, nine nodes will be used in a small area (3x3 meters).

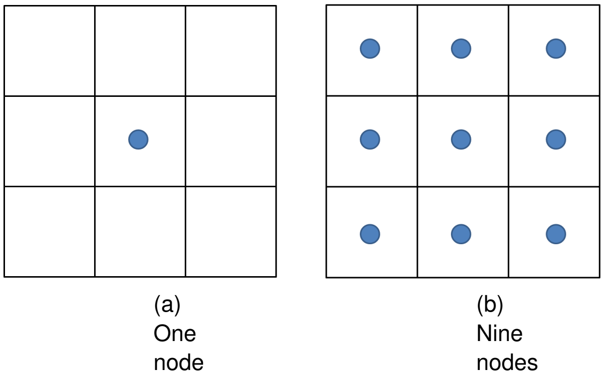

(a) One node (b) Nine nodes

Figure 3.12: Scalability

Measurement Method Also the method to measure this metric is comparable with the previous method. Two scenarios with exactly the same configuration are executed with the only difference the amount of localized nodes.

3.1.4.7 Repeatability

Table 3.17: Repeatability: summary

NAME	Repeatability
TYPE	Derived performance
SUBMETRICS	
SHORT DESCRIPTION	The repeatability is evaluated by identifying the difference in primary metrics to multiple repetitions of a same benchmarking scenario.

Description This metric defines if a solution is stable or not. Therefore the solution will be reinstalled multiple times and check the variation in the accuracy. (The whole solution must be degraded and rebuild.)

Measurement Method The measurement of a certain metric will be executed multiple times on the same testbed under the same conditions. Then, the deviation can be calculated.

3.1.4.8 Functional Metrics

Functional metrics are used to evaluate the non-performance related attributes of a localization solution. Functional metrics are listed for completeness but they are not the main focus of EVARILOS.

Technology Type

Description The metric “solution requirements” can be split up in two parts. A distinction can be made between the following items:

- The **algorithms requirements** include the computational complexity, the memory requirements, server(s), etc.
- A second item are the **technology requirements**. The used technology influences the type of hardware that can be used, e.g. a tag, smartphone, etc.

The summary can be found in Table 3.18.

Table 3.18: Technology type: summary

NAME	Solution requirements
TYPE	Functional
SUBMETRICS	
SHORT DESCRIPTION	The requirements influence the type of hardware that can be used to deploy a localization solution.

Measurement Method A classification of the requirements can be made, for example by differentiating by sensors, smartphones, pc’s and servers.

Measurement Unit There is no explicit unit for this metric. There are different classes and each solution gets a score from 1 to 10, depending on the class the solution belongs to.

Open Source

Description This metric handles about the license of the source code. Depending on the restrictiveness of the license of the software, different scores can be assigned. The summary of this metric can be found in Table 3.19.

Table 3.19: Open source: summary

NAME	Open source
TYPE	Functional
SUBMETRICS	
SHORT DESCRIPTION	Is the software open source available?

3.1.4.9 Deployment Metrics

Deployment metrics are used to evaluate the complexity of installing a localization solution. Deployment metrics are listed for completeness but they are not the main focus of EVARILOS.

Set-up Overhead

Table 3.20: Set-up overhead: summary

NAME	Set-up overhead
TYPE	Deployment
SUBMETRICS	• Physical installation • Configuration • Replacement time
SHORT DESCRIPTION	The set-up overhead includes different factors that indicate the complexity of installation.

Physical Installation

Description The physical installation measures the time that is needed to install the complete system.

Measurement Method Assuming the necessary power plugs are available, the time is measured from the moment installation of the localization hardware is started until all physical components are installed correctly. This time is multiplied by the total number of persons installing the solution.

Measurement Unit Time is expressed in `seconds (s)`.

Configuration

Description The configuration is a combination of multiple parameters concerning the configuration complexity, e.g. does the solution require fingerprinting or not? The answers to these questions influence the complexity of the configuration in the set-up overhead.

Measurement Method To “measure” this metric, a questionnaire is used that utilizes multiple choice of answers. Each answer will have a certain cost value.

- Does the solution require fingerprinting or not?
- Do you need to manually enter coordinates?
- How are these nodes mounted on the ceiling?
- How are the central components installed?
- Are their limitations to the locations where the anchor points can be installed?

Measurement Unit The “configuration complexity” is a number without an explicit unit. This value has no limit. It starts with 0. The lower the configuration complexity is, the better the solution is.

Replacement Time

Description Finally, the replacement time represents the autonomy of the system.

Measurement Method This metric is expressed in `time`. The time is measured how long the system can typically operate without the intervention of a human being: e.g. how often a recalibration is necessary.

Measurement Unit `Time` is expressed in `seconds` (s).

Hardware Cost

In the deployment metrics, a distinction between the set-up overhead and hardware cost is made. The set-up overhead expresses the cost in time. However, the hardware cost expresses the financial costs of the system. A summary can be found in Table 3.21.

Table 3.21: Hardware cost: summary

NAME	Hardware cost
TYPE	Deployment
SUBMETRICS	• Fixed nodes • Localized nodes
SHORT DESCRIPTION	The financial value of the complete system is evaluated in this metric.

Infrastructure Nodes

Description The financial cost of the installation and fixed nodes is a one-time cost that includes the amount and price tag of the fixed nodes, as well as the server costs.

Measurement Method EVARILOS estimates the number of devices multiplied by a certain reference cost. There are different classes and each solution gets a score from 1 to 10, depending on the class the solution belongs to.

Measurement Unit The financial values are expressed in `Euro` (€) and the classification number has no unit.

Localized Nodes

Description The price tag of the localized nodes is evaluated separately. Especially in rough conditions, a localized node will need frequent replacements. An example is underground mining: mineworkers can easily damage or lose a tag, so the use of expensive localized nodes is in this case not desired.

Measurement Method EVARILOS estimates the number of devices multiplied by a certain reference cost. There are different classes and each solution gets a score from 1 to 10, depending on the class the solution belongs to.

Measurement Unit The financial values are expressed in `Euro` (€) and the classification number has no unit.

3.2 List of EVARILOS Benchmarks

Finally, this section describes the list of the benchmarks that are defined by the EVARILOS consortium. These benchmarks have been used and evaluated throughout the previous EVARILOS deliverables, most notably D2.2. "Experiments Without Interference" and D2.3. "Experiments With Interference". The benchmarks below give very concrete specifications for the settings that should be used for an experiment. Although other configuration choices could be motivated, their main purpose is (i) to provide default values that make sense for many research purposes, and (ii) to obtain a clear and accurate description of experiments that were performed. Experimenters can refer to those benchmarks to create benchmarking scenarios or when reporting on their results, thereby allowing better comparability of the conditions under which localization solutions were evaluated.

3.2.1 Benchmark 1: No Interference

This benchmark is used to create reference scenarios for evaluating an indoor localization solution under well-controlled conditions.

3.2.1.1 Operating Conditions

A basic requirement for this benchmark is that the environment should be under control of the experimenter. The experimenter should take care to:

- Minimize interference. Preferable, the environment should be shielded and the experiments should be performed when no other wireless activities are present. Interference levels have to be monitored before and after each evaluation: when the interference level exceeds a SUT-dependent threshold, the measurement should be discarded and a new evaluation should be performed at this location.
- Stable environment. To promote repeatability, no human activities (besides the one necessary for the evaluation) should be present.
- Typical operational conditions. The experiment should be performed in typical operational conditions in terms of temperature and humidity.

3.2.1.2 Selection of Evaluation Points

To select the evaluation points, an orthogonal grid is created according to the following guidelines.

- To prevent a bias in the selection of the grid coordinates, the interval between the different grid coordinates should be evenly distributed. However, for non-square buildings, the interval between the X coordinates is allowed to differ from the interval between the Y coordinates.
- In addition, the grid should be sufficiently dense.
  - The maximum distance between grid points is 5 meter for both X and Y dimensions.
  - The interval between the grid coordinates in X dimension should be equal to or lower than $\lceil X\_dimension_{area}/5 \rceil$.
  - Similarly, the interval between the grid coordinates in Y dimension should be equal to or lower than $\lceil Y\_dimension_{area}/5 \rceil$.
  - For non-square buildings, multiple grids can be created over the area.

The resulting grid is uniformly distributed over the area and consists of at least 25 grid points, though more are preferred.

Next, a minimum of 20 evaluation points is selected on the grid according to the following guidelines.

- Two sampling methods are allowed.
  - Uniform sampling over all the grid points.
  - Latin Hypercube Sampling to ensure that there is at least one sample in each row and each column.
- When Latin Hypercube Sampling results in less than 20 evaluation points, additional evaluation points on the grid can be added to reach the minimum amount of evaluation points, although in this case starting with a more dense grid is the preferred option.
- For evaluation points that are not accessible, instead the nearest accessible location is selected.

Although more complex to calculate than the grid approach, other randomly distributed uniform distributions are also allowed, including Uniform Jitter Sampling, Best Candidate Sampling or Poisson Disk Sampling [12]. Finally, it is worth noting that the evaluation points are not allowed to overlap with the training points that were used, e.g. for taking fingerprints.

3.2.1.3 Evaluation Procedure

The evaluation procedure takes place according to the following rules.

- The device to be localized is moved to one of the evaluation locations.
- The device stays stationary as long as the time needed for each measurement.
- Devices are physically disabled when moving to a different evaluation point to ensure no other tracking information (dead reckoning, step counters, etc.) is used.
- All selected evaluation points are visited once and no evaluation points are revisited.
- SUT's that use IMU will not be penalized, but the methodology is primarily designed for discrete points.

3.2.1.4 Evaluation Criteria

The output of this benchmark should, as a minimum, include the following metrics:

- Point accuracy (mean, standard deviation, minimum and maximum values).
- Latency (mean, standard deviation, minimum and maximum values).
- Energy efficiency of the localized devices.

3.2.2 Benchmark 2: IEEE 802.11 Interference

This benchmark is used to create reference scenarios for investigating the influence of IEEE 802.11 (WiFi) traffic on the performance of localization solutions. The benchmark is very similar to benchmark 1 and only differs in its operating conditions.

3.2.2.1 Operating Conditions

During the evaluation stationary evaluation phase, IEEE 802.11 traffic is generated with the following characteristics.

- Interference is generated using IEEE 802.11g interfaces.
- The transmission power is set to 20 dBm and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used.
- Each interferer transmits a continuous data stream of 24 Mbit/s during each evaluation phase.
- The transmission frequency should overlap maximally with the frequency used by the solution under test. If the solution under test uses multiple frequencies, it should be configured to overlap maximally with the frequency that is used initially.
- The number of transmitting interfering devices is equal to the evaluation area (in square meters) divided by 1000, rounded up.
- Transmitters are spread out uniformly over the evaluation area.
- Between two evaluation points, the experiment must be monitored for outside interference, discarding the results if outside interference was detected before or after the experiment.

Following the above interference specifications the experimenter can generate the interference conditions in which, barring extreme area shapes, all evaluation points are influenced by interference. In addition, the density of the transmitters is such that the area will consist of both strongly and weakly interfered evaluation points.

3.2.3 Benchmark 3: IEEE 802.15.4 Interference

This benchmark is used to create reference scenarios for investigating the influence of IEEE 802.15.4 traffic on the performance of indoor localization solutions. The benchmark is similar to the benchmark 1, with the difference in the operating conditions.

3.2.3.1 Operating Conditions

During the evaluation stationary evaluation phase, IEEE 802.15.4 traffic is generated with the following characteristics.

- Interference is generated using IEEE 802.15.4 devices.
- The transmission power is set to 0 dBm and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used.
- Each interferer transmits a continuous data stream of 256 Kbit/s during each evaluation phase.
- The transmission frequency should overlap maximally with the frequency used by the SUT. If the SUT uses multiple frequencies, it should be configured to overlap maximally with the frequency that is used initially.
- The number of transmitting interfering devices is equal to the evaluation area (in square meters) divided by 100, rounded up.
- Transmitters are spread out uniformly over the evaluation area.
- Between two evaluation points, the experiment must be monitored for outside interference, discarding the results if outside interference was detected before or after the experiment.

Following the above interference specifications the experimenter can generate the interference conditions in which, barring extreme area shapes, all evaluation points are influenced by interference. In addition, the density of the transmitters is such that the area will consist of both strongly and weakly interfered evaluation points.

3.2.4 Benchmark 4: Synthetic Interference

This benchmark is used to create reference scenarios for investigating the influence of synthetic interference on the performance of indoor localization solutions. The main difference with the previous benchmarks is that the generated interference has no provisions for detecting and avoiding collisions, such as CSMA/CA. The benchmark is very similar to the benchmark 1, only the operating conditions are different.

3.2.4.1 Operating Conditions

During the evaluation stationary evaluation phase, synthetic traffic is generated with the following characteristics.

- Interference is generated using a power envelope of IEEE 802.11b/g modulated signal (20 MHz bandwidth).
- The transmission power is set to 20 dBm, no CSMA/CA is used.
- Each interferer transmits a continuous jamming signal during each evaluation phase.
- The transmission frequency should overlap maximally with the frequency used by the solution under test. If the solution under test uses multiple frequencies, it should be configured to overlap maximally with the frequency that is used initially.
- The number of transmitting interfering devices is equal to the evaluation area (in square meters) divided by 1000, rounded up.
- Transmitters are spread out uniformly over the evaluation area.
- Between two evaluation phases, the experiment must be monitored for outside interference, discarding the results if outside interference was detected before or after the experiment.

Following the above interference specifications the experimenter can generate the interference conditions in which, barring extreme area shapes, all evaluation points are influenced by interference. In addition, the density of the transmitters is such that the area will consist of both strongly and weakly interfered evaluation points.

Chapter 4

Workflows

The previous chapter described a set of well-defined benchmarks, i.e. general methodologies for creating a benchmark scenario used to evaluate a localization solution. *A workflow describes how to combine the output from multiple benchmark scenarios (i.e. instantiated benchmarks)*. This can be used for the calculation of derived performance metrics on the one hand. On the other hand, workflows are also the solution to go beyond the black-box testing. Workflows answer the question of how the performance metrics are influenced when individual aspects are varied while all other parameters are kept the same. As such, a workflow typically consists of one or more input benchmarking scenarios, and as output a number of derived performance metrics. Workflows can be used (i) to gain *insights in how the behavior of the localization solution changes under different conditions*, (ii) to *find optimal values* of a solution under test or (iii) for *advanced score calculation that takes into account the behavior of a system under multiple conditions*.

An example of the first is given below. To test the predictive value of the bechmarking scenario results in different environments, one has to perform many repetitions of a single benchmarking scenario, trying to isolate the impact of different environments on the benchmarking result.

1. Keep SUT parameters unchanged.
2. Iterate over several different environments and perform the benchmarking experiments.
3. Process the results from the individual experiments, aiming to establish correlation between features in the environment and the results.
4. Use the established correlation hints to predict performance of the given SUT in a new environment that shares elementary characteristics with the set of evaluated environments.

An example of the second is given below. To give suggestions about the optimal parametrization of a given SUT, one can utilize the benchmark in the following way:

1. Keep the evaluation environment and configuration (number of evaluation points, etc.) fixed.
2. Perform a benchmarking experiment.
3. Vary the SUT parametrization in a limited set of dimensions (number of anchor points, number of collected RSSI packets per training point, etc.).
4. Process the results from the individual experiment, aiming to establish a correlation between the scores and the parameter values of the different SUT sub-variants.

5. Use the established correlation hints to suggest optimal parametrization heuristics for environments similar to the one for which the experiment batch has been performed.

Gaining these insights requires a large number of experiments, the exact number will vary on the complexity of the research question or optimization. Although statistical conclusions be made after performing enough experiments (e.g. utilizing a Student's t-Test), in practice most experiments do not have the time and resources to perform such a number of experiments. A trade-off between usability and statistical correctness was made in the definition of the workflows, heavily favouring usability. As such, the workflows below are used mainly to gain insight about the impact of changing operational conditions, but can not be used to derive general conclusions. If more statistically relevant conclusions are required, we encourage increasing the number of experiments significantly. Typically, design of experiment approaches that find the optimum number of experiment settings and their suggested values can be used to this end.

4.1 Workflow 1: Interference Sensitivity

Interference sensitivity expresses the impact a certain interference type has on the accuracy and the responsiveness of a localization solution. For small number of experiments, the derived metric for interference sensitivity in Section 3.1.4.3 can be used. For larger number of repetitions, the interference sensitivity can also be evaluated using the following modified metric.

Input values

The workflow combines the outputs from an experiment implementing benchmark 1, as defined in Section 3.2.1, with any of the experiments implementing benchmarks x (with x = 2, 3 or 4) as defined in sections 3.2.2 through 3.2.4.

Output Values

In the assumption that we have N experiments in the reference scenario (ref_1, ref_2,...ref_N), and M experiments from an interference scenario (int_1, int_2, ..., int_M). The Interference Impact (II) for multiple performance metrics is calculated according to the following equations:

$$II_{Point\ Accuracy} = max\left(\frac{mean\,[\,PA_{int_1}\ ,\ ...\ ,\ PA_{int_M}\,] - mean\,[\,PA_{ref_1}\ ,\ ...\ ,\ PA_{ref_N}\,]}{mean\,[\,PA_{ref_1}\ ,\ ...\ ,\ PA_{ref_N}\,]}\ ,\ 0\right) \tag{4.1}$$

$$II_{Room\ Accuracy} = max\left(\frac{mean\,[\,RA_{ref_1}\ ,\ ...\ ,\ RA_{ref_N}\,] - mean\,[\,RA_{int_1}\ ,\ ...\ ,\ RA_{int_M}\,]}{mean\,[\,RA_{ref_1}\ ,\ ...\ ,\ RA_{ref_N}\,]}\ ,\ 0\right) \tag{4.2}$$

$$II_{Latency} = max\left(\frac{mean\,[\,RT_{int_1}\ ,\ ...\ ,\ RT_{int_M}\,] - mean\,[\,RT_{ref_1}\ ,\ ...\ ,\ RT_{ref_N}\,]}{mean\,[\,RT_{ref_1}\ ,\ ...\ ,\ RT_{ref_N}\,]}\ ,\ 0\right) \tag{4.3}$$

The equations calculate the ratio decrease of the accuracy and the ratio increase of the latency in the scenarios with interference, compared to the reference scenario. Higher values indicate that interference has a higher impact. In all three equations, if the result is negative, the result is capped to 0. The reasons for negative interference impacts that can occur, although rarely, i.e. better performance in the benchmarking scenario with controlled interference in comparison to the reference scenario, are randomness in the wireless environment combined with a low impact of the interference.

Finally, the overall interference impact of a given SUT is the average of the interference sensitivity of all metrics. Higher values indicate higher influence of interference on an indoor localization solution.

$$\overline{II} = \frac{II_{Point\ Accuracy} + II_{Room\ Accuracy} + II_{Latency}}{3} \tag{4.4}$$

4.2 Workflow 2: Environmental Sensitivity

The workflow combines the outputs from at least two instantiations of benchmark 1 in different environments. The output expresses how sensitive the solution is to different environments. For small number of experiments, the derived metric for environmental sensitivity in Section 3.1.4.3 can be used. For larger number of repetitions, the environmental sensitivity can also be evaluated using the following modified metric.

Output Values

Assuming we have N experiments, the Environment Impact (EI) for multiple performance metrics is calculated according to the following equations:

$$EI_{Point\ Accuracy} = std\ [avg\ PA_1\ ,\ avg\ PA_2\ ,\ ...\ ,\ avg\ PA_N] \tag{4.5}$$

$$EI_{Room\ Accuracy} = std\ [avg\ RA_1\ ,\ avg\ RA_2\ ,\ ...\ ,\ avg\ RA_N] \tag{4.6}$$

$$EI_{Latency} = std\ [avg\ RT_1\ ,\ avg\ RT_2\ ,\ ...\ ,\ avg\ RT_N] \tag{4.7}$$

or more general

$$EI_x = std\ [avg\ x_1\ ,\ avg\ x_2\ ,\ ...\ ,\ avg\ x_N] \tag{4.8}$$

The equations calculate the standard deviation of the averages of the experiments. Finally, the overall environment impact of a given SUT is the average of the interference sensitivity of all metrics. Higher values indicate higher influence of the environment on the performance of the indoor localization solution and as such give an indication of the reproducibility of a SUT in different environments.

$$\overline{EI} = \frac{EI_{Point\ Accuracy} + EI_{Room\ Accuracy} + EI_{Latency}}{3} \tag{4.9}$$

4.3 Workflow 3: Repeatability

The workflow combines the outputs from at least two similar instantiations of benchmark 1 in the same environment. The output expresses how sensitive the solution is to multiple repetitions of the same experiment. For small number of experiments, the derived metric for repeatability in Section 3.1.4.3 can be used. For larger number of repetitions, the repeatability can also be evaluated using the following modified metric.

Output Values

Assuming we have N experiments, the repeatability R for multiple performance metrics is calculated according to the following equations:

$$R_{Point\ Accuracy} = std\ [avg\ PA_1\ ,\ avg\ PA_2\ ,\ ...\ ,\ avg\ PA_N] \tag{4.10}$$

$$R_{Room\ Accuracy} = std\ [avg\ RA_1\ ,\ avg\ RA_2\ ,\ ...\ ,\ avg\ RA_N] \tag{4.11}$$

$$R_{Latency} = std\ [avg\ RT_1\ ,\ avg\ RT_2\ ,\ ...\ ,\ avg\ RT_N] \tag{4.12}$$

or more general

$$R_x = std\ [avg\ x_1\ ,\ avg\ x_2\ ,\ ...\ ,\ avg\ x_N] \tag{4.13}$$

The equations calculate the standard deviation of the averages of the experiments. Finally, the overall repeatability of a given benchmarking scenario is the average of the repeatability of all metrics. Higher values indicate low repeatability scores, and indicate that either the solution is inherently unstable, or that the benchmarking scenario can not be relied upon due to external influences, too limited amount of evaluation points, etc.

$$\overline{R} = \frac{R_{Point\ Accuracy} + R_{Room\ Accuracy} + R_{Latency}}{3} \tag{4.14}$$

4.4 Workflow 4: Anchor Nodes Selection

The workflow investigates the influence of the anchor nodes selection. It combines the outputs from at least two instantiations of reference benchmark 1 in which the anchor nodes are selected differently (i.e., a higher or lower number of anchor nodes, or anchor nodes chosen in different locations). The output expresses how sensitive the solution is to the choice of anchor nodes.

Output Values

Assuming we have N experiments, the Anchor Node (AN) influence for multiple performance metrics is calculated according to the following equations:

$$AN_{Point\ Accuracy} = std\ [avg\ PA_1\ ,\ avg\ PA_2\ ,\ ...\ ,\ avg\ PA_N] \tag{4.15}$$

$$AN_{Room\ Accuracy} = std\ [avg\ RA_1\ ,\ avg\ RA_2\ ,\ ...\ ,\ avg\ RA_N] \tag{4.16}$$

$$AN_{Latency} = std\ [avg\ RT_1\ ,\ avg\ RT_2\ ,\ ...\ ,\ avg\ RT_N] \tag{4.17}$$

or more general

$$AN_x = std\ [avg\ x_1\ ,\ avg\ x_2\ ,\ ...\ ,\ avg\ x_N] \tag{4.18}$$

The equations calculate the standard deviation of the averages of the experiments. Finally, the overall influence of anchor nodes selection is the average of the influence of anchor nodes selection for all metrics.

$$\overline{AN} = \frac{AN_{Point\ Accuracy} + AN_{Room\ Accuracy} + AN_{Latency}}{3} \tag{4.19}$$

Chapter 5

Final Score Calculation

The benchmarks and workflows described in the previous chapters give several output parameters that give a good overview of the behavior and performance of evaluated localization solutions. As such, objective ranking is already possible based on the output metrics of benchmarks and workflows. However, when confronted with a wide range of available output metrics, human nature typically tries to abstract and simplify the available data by trying to assign a final definitive score value to the overall solution. Score calculations can be used to create a single ranking of localization solutions based on multiple criteria, thereby enabling a quick comparison between a large number of localization solutions. Such a final score needs to include aspects describing a wide range of performance metrics and, out of necessity, abstracts away a significant amount of information. As such, a score value can not be used to gain additional insight in the behavior of the localization solution: it merely offers a convenient way to make quick quantitative comparisons between multiple solutions.

Since different application domains have strongly different requirements, another disadvantage of using scores is that they are typically application or even user dependent. Indeed, due to the wide range of potential application-specific interests and trade-offs, it is impossible to define a single score calculation method that is both simple and will satisfy the constraints of all potential application domains. As such, instead of providing a single generic score calculation approach, we discuss several score calculation options that will be frequently be relevant, and leave it to the reader to come up with alternatives if the presented approaches do not fit exactly the targeted comparison method.

5.1 Simple Score Calculation

A first method is to assign overall scores is the following. The calculation of an overall score for each metric is done according to a linear function that is defined by specifying minimum and maximum acceptable value for the metric. Furthermore, multiple metric dependent scores are combined by using weighting factors that define each metric's importance for a given use case.

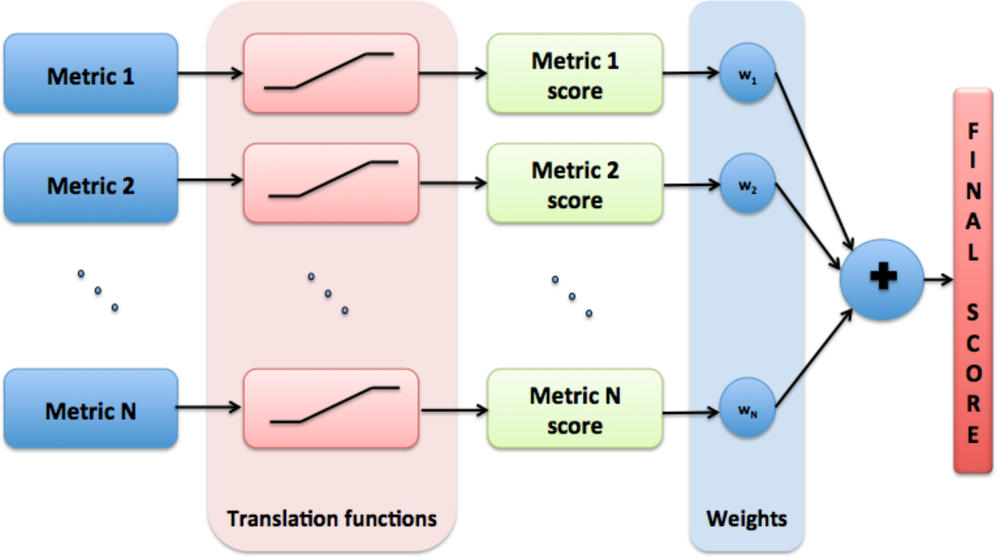


Figure 5.1: Calculation of the final score

In general, the linear translation function for calculating the score of each particular metric is given in Formula 5.1, where score can vary from 0 to 10. Minimum and maximum acceptable values are defined with M_{min} and M_{max}, respectively. M_{min} can be bigger than M_{max}, e.g. in defining the acceptable point accuracy values one can discuss about acceptable localization error margins. Here M_{min} is the biggest acceptable error, while M_{max} is the desired average localization error.

Note that the formula is reversed for metrics in which a higher value represents a more desirable result, e.g.: for point accuracy lower values are better. Also, more complex functions than a linear one could be envisioned. The main advantage of this approach is the simplicity with which multiple metrics are combined to form an overall score.

$$Score = \max\left(0, \min\left(10, 10\frac{m - M_{min}}{M_{max} - M_{min}}\right)\right) \tag{5.1}$$

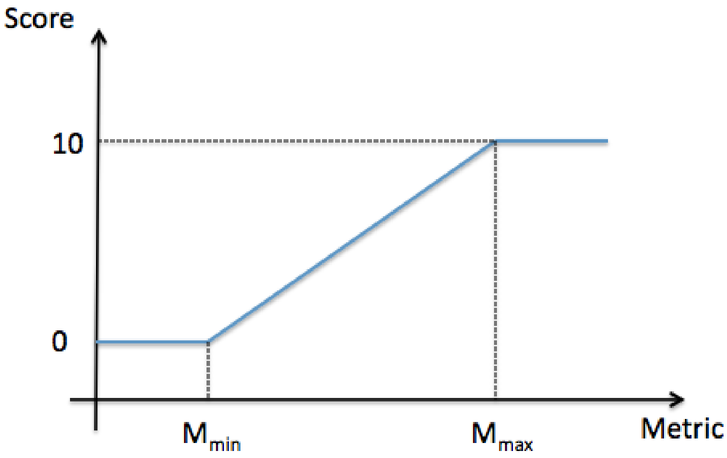


Figure 5.2: Linear translation function for each metric in case when $M_{min} < M_{max}$

The min and max score can vary depending on the use case. As an example, we demonstrate the minimum and maximum score thresholds for two specific use cases related to the healthcare and underground mining. The use cases are taken from the report [9] on the definition and setup of the validation scenarios in healthcare and underground mining, which is the result of Task 4.1. and Task 4.2 of the EVARILOS project. For the healthcare environment we consider a use case in which different medical devices or persons (nurses & patients etc.) have to be localized. For the mining environment we consider a similar scenario, i.e. localization of the machinery used in mines. The weights, and minimum and maximum values for each metrics are presented in the Table 5.1.

Table 5.1: Metric scores and weight factors for the healthcare and mining scenarios

Metric	M_{min}	M_{max}	w
Small healthcare scenario			
Point accuracy [m]	4	2	0.2
Room accuracy [%]	80	100	0.4
Latency [ms]	3 000	20	0.3
Energy efficiency [mW]	5 000	20	0.1
Big healthcare scenario			
Point accuracy [m]	6	3	0.2
Room accuracy [%]	80	100	0.5
Latency [ms]	3 000	100	0.2
Energy efficiency [mW]	5 000	20	0.1
Big open space mining scenario			
Point accuracy [m]	12	5	0.7
Latency [ms]	30 000	1 000	0.2
Energy efficiency [mW]	20 000	100	0.1

5.2 Constrained Score Calculation

An optimization of the above technique consists of including a “knock-out criteria”. The same calculation is still used, but if a certain metric exceeds a given threshold, the solution can be “knocked out” and is marked as “unacceptable”. This prevents situations in which solutions that have one or two unacceptable metrics, but have low weights for these metrics, resting in high scores. This knock out criteria will be applied on the average value of each metric.

5.3 Multi-variate Score Calculation

In some cases, a returned location estimate is only useful if it falls within the acceptable range of two or more metrics. For example, for an urgent alarm, a location estimate might be required that is both accurate and with low delay. These use cases can be accommodated by combining multiple metrics before translating both into one score. These types of scores better represent the inherent trade-off between multiple metrics. The latter are referred to as multi-variate approaches: multi-variate statistics is a form of statistics encompassing the simultaneous observation and analysis of more than one outcome variable.

A conceptual way to visualize this is to display multiple metrics on a single graph that is divided into different quadrants. Each quadrant could represent a pre-specified score. An example is shown in Figure 5.3: if the location estimate falls within the red zone, the result does not satisfy the application requirements. In the orange zones, the solution can be defined as “weak”. Only the location estimates that have both a response delay and an accuracy below a predefined threshold are given the highest score. An example is given in Figure 5.4. The score calculation can be done as in Equation 5.2.

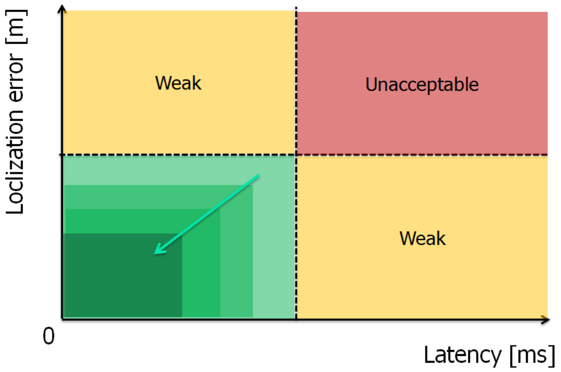


Figure 5.3: Dividing a graph with two metrics in multiple quadrants

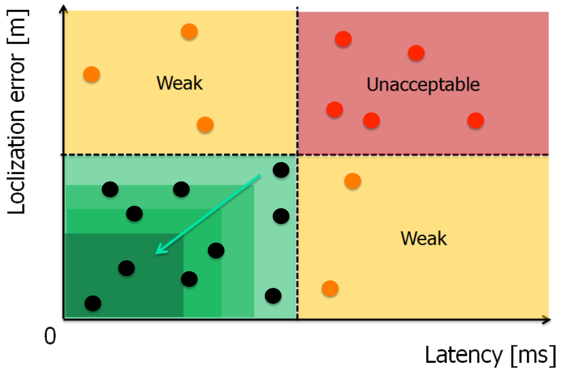


Figure 5.4: Dividing a graph with two metrics in multiple quadrants: applied example

$$
\begin{aligned}
Score_{acceptable} &= \frac{\#\text{ acceptable estimates}}{\#\text{ all estimates}} = \frac{10}{20} = 50\% \\
Score_{weak\ in\ accuracy} &= \frac{\#\text{ estimates with weak accuracy}}{\#\text{ all estimates}} = \frac{3}{20} = 15\% \\
Score_{weak\ in\ latency} &= \frac{\#\text{ estimates with weak latency}}{\#\text{ all estimates}} = \frac{2}{20} = 10\% \\
Score_{unacceptable} &= \frac{\#\text{ unacceptable estimates}}{\#\text{ all estimates}} = \frac{5}{20} = 25\%
\end{aligned}
\tag{5.2}
$$

The above example combines the averages of two performance metrics for calculating final scores. Multi-variate approaches can also consider the combined performance metrics at each evaluation point and as such also take into account the distribution of multiple metrics. Using the EVARILOS Benchmarking Suite (Deliverable 2.5 [8]) these advanced calculations are also possible: EVARILOS makes available the results from its own localization solutions (Deliverables D2.2 [6] and D2.3 [7]), as well as of those solutions that participated in the EVARILOS Open Challenge [10]. Each of these datasets also has its associated experiment configuration settings, allowing detailed analysis of not only the performance but also the conditions in which the solutions were evaluated. As such, it is possible to define scoring approaches that incorporate the behavior over time (temporal aspects), behavior over space (spatial effects) or even how different performance metrics are correlated to each other.

Chapter 6

Conclusions

RF-based indoor localization solutions are increasingly popular with researchers that aim to provide more accurate and robust solutions. The multiplicity of RF-based indoor localization solutions makes their evaluation an indispensable part of future Internet. However no unified scheme has been devised for evaluation of these solutions and their robustness against various parameters. To remedy this, the EVARILOS Benchmarking Handbook is created in order to objectively evaluate and compare different indoor localization solutions.

The EVARILOS benchmarking approach is based on *system-level testing*, i.e. evaluating the localization solution as a whole, without differential treatment of the individual components comprising the system (also referred to as “black box testing”). By concentrating on the performance of the system on the highest functional level, the system testing comes closest to the interests of the end-users of the localization systems, who are mostly interested whether the system as a whole meets their specific requirements.

To create a foundation for comparative evaluation of different localization solutions and their ranking according to a use case specific scoring keys, a *generic benchmarking methodology* was described. The methodology consists of three major steps: a measurement phase during which the statistics are stored, an evaluation phase during which the metrics are calculated based on the earlier measurements, and finally an optional scoring phase that can be used to create an application-dependent ranking of localization solutions. In addition, an overview is given of the used the terminology, together with the definition of a consistent and unambiguous set of evaluation metrics that can be used to evaluate localization solutions.

The deliverable also includes the part of the EVARILOS Benchmarking Handbook that details how the previous methodology is used to create *well-defined benchmarks*. Benchmarks describe a methodology for creating an evaluation scenario for indoor localization solutions. They can be used to create benchmarking scenarios that give very concrete specifications for the settings that should be used for an experiment. Their main purpose is to provide default values for scenarios that will result in setups that do not have immediate flaws and will result in comparable outcomes. A list of predefined benchmarks that can be instantiated in multiple environments is provided.

EVARILOS also investigates how the presence of interference impacts current localization solutions. To answer the question on how performance metrics change in different configurations, *a set of workflows was provided, describing how to combine the output from multiple benchmark scenarios to calculate derived performance metrics*. A workflow is build on top of elementary processes and can be used (i) to gain insight in how the behavior of the localization solution changes in different conditions, (ii) for advanced score calculation that takes into account the behavior of a solution in multiple conditions. In other words, workflows can be useful in parametrization of algorithms. By providing a well-described set of workflows and their associated calculation of derived metrics, objective comparability between different results is further improved.

The performance metrics of the benchmarks and workflows can be used for ranking of solutions. For those that are interested in obtaining a single final score - out of necessity abstracting away many details of the performance evaluation of the solution - we have provided several options for transforming the evaluation metrics of benchmarks and workflows into a final application dependent evaluation score. Due to the open nature of the EVARILOS data repository, experimenters that want alternative score calculation options can easily utilize the available experiment data for custom score calculations.

In summary, this document offer several options for accurate and objective evaluation of indoor localization solutions, as well as well-described benchmarks to prevent experimenters from performing experiments that are not scientifically complete, thereby offering comparability that is significantly beyond what is currently found in the state-of-the-art scientific literature.

Appendix A

Application Domains

A.1 Introduction

The solutions developed in the EVARILOS project are designed to be applicable for any application domain. The performance of different localization solutions are given in the form of metrics. By adjusting the weighting scores of the individual metrics, the suitability of a specific localization solution for a specific application domain can be deducted. In the EVARILOS project, two specific application domains (‘healthcare’ and ‘underground mining’) are described that correspond to the respective activities of the two involved SME’s: Televic Healthcare and AdvanticSys. The application domains are described below with additional details given in [9].

A.2 Application Domain 1: Healthcare

A.2.1 Introduction

In recent years the complexity in nursing organizations has been increasing due to societal factors such as the increase of the care unit size, the increase of specialized care and the lack of nurse staffing which requires a more efficient use of resources. In addition to these inherent factors, a further increase of complexity is due to the high amount of technology that is being introduced for the staff (e.g. medical equipment, pagers, alert redirecting and electronic medical records) as well as for the environment (e.g. building automation for energy control and comfort functions for the patient). In future years these complexity trends will continue due to upcoming technologies, such as location aware services, and computerized decision support systems, and an aging society, which translates into an increasing need for care and a decreasing number of available staffing.

A.2.2 Healthcare Demands Wireless

During the last couple of years, the demand for wireless systems has increased significantly. Offering wireless nurse call (patient, nurses, assets are equipped with a mobile tag) has the following advantages:

- The patient feels more free because holding the call button does not invoke cables
- The patient feels more secure as an alarm can be launched at any place at any time

Wireless technology further allows offering location based services. It is clear that localization is the enabling technology that creates new applications, services and added value for patients, staff and the entire healthcare organization. When looking to the market, the demand for additional application such as wandering detection and access control is very large. In this case patients are allowed to have more freedom without losing control of their safety. Embarrassing situations where patient are tied to their bed can be avoided.

A wireless nurse call system with localization is one of the main drivers for a shift from a room oriented nurse call system to a person oriented nurse call system.

- In a room oriented system, a patient can push a button on a fixed infrastructure and (a group of) nurses will receive a message that an alarm was triggered in room 512.
- In a person oriented system, the patient will take a central position and the care giving activities will be optimized to obtain the optimal treatment. In this case, the most “appropriate” nurse will receive an alarm telling that John needs help in the corridor. In this system is possible to handle calls on a much more efficient way if also the location of the staff is known. On top of this the entire organization will benefit from adopting localization technology especially when also assets can be localized.

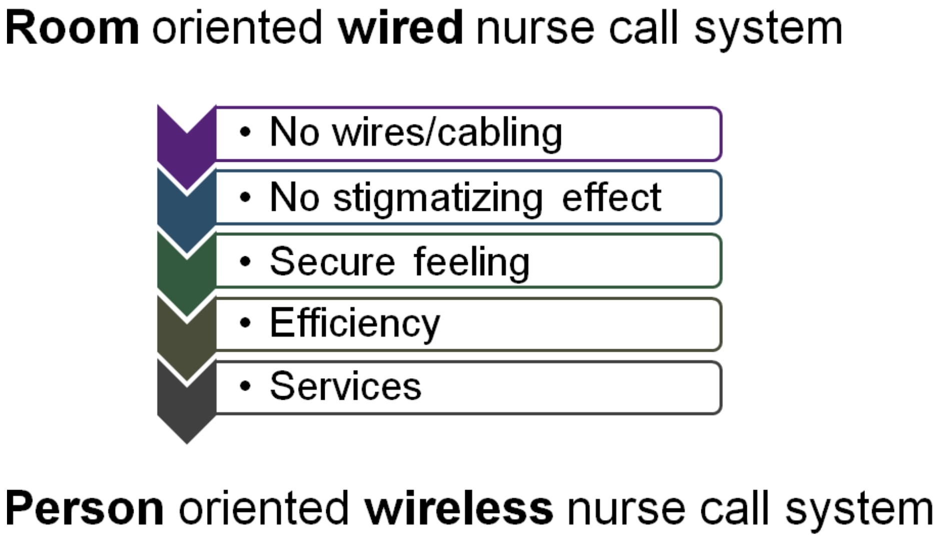


Figure A.1: Person oriented wireless nurse call system

As already mentioned, there exist a large interest for such products and solutions. However the adoption is still rather low because

- The localization accuracy is insufficient. Typically room level accuracy is what matters which can be very challenging close to the walls
- The solution is too expensive. The costs includes
  - Cost of the tags
  - Infrastructure: the number of access points/beacons
  - The installation procedure (additional cabling, calibration time...)
  - Software + user interface
  - Other license costs
- The life time of the tag is too short

- The system should be very easy to use: nurses have no time to find out how it works
- The solution is tailored to one specific scenario but cannot support others in a proper way (typical use cases are discussed further)
- The size of the tag is too large which makes it not nice to wear or too stigmatizing (feeling watched)

A.2.3 Basic Nurse Call Scenarios

A.2.3.1 Use Case 1: Locating a Patient

When a patient issues an emergency call, the nurse has to know the location of that patient in order to respond correctly. When the alarm buttons are part of a wired installation, the location is inherent to the installation. In the wireless case however, the basic question “where did the call originate” is less obvious since wireless signals are attenuated by the walls, floors and ceilings, but they can travel through them. Depending on the actual position of the patient when he issues the emergency call, the location has to be known in more or less detail.

In case he’s inside a patient room, the number of the room is sufficient to identify his location. In case of a multi-bed room, the bed number could be further relevant information, however the notion that the location of the patient is inside a room is sufficient information to act upon the emergency call.

If the alarm is issued in a long corridor however, the notion that he is located inside the corridor is no longer adequate: the location information has to contain more details (e.g. located in corridor one, near room 142). Especially for large hospitals corridors can stretch out quite far, e.g. corridors that are used to interconnect several buildings on a hospital campus.

In a third case, a large public room (like the hospital cafeteria), the location information should indicate a more accurate region of the large room (e.g. a quadrant or a table number in case the tables are numbered).

A.2.3.2 Use Case 2: Emergency Call by Patient

Reacting on emergency situations is critical in hospital environments. Also in other applications this concept exists: e.g. a gas alarm in a network of gas detectors in an underground garage, reacting on a lamp defect in an emergency lighting setting, etc. This use case is characterized by a very simple device, the patient device, being able to set off an alarm in a different part of the network. The patient device not only needs to have very low complexity, it must also operate at low power. Furthermore, the reliability of the emergency call system must be extremely high, i.e. the probability of the button press on the patient device triggering an alarm must be extremely high. Combined with the relative low cost requirement for this type of device and hence its required simplicity, special measures need to be taken to ascertain the robustness of this use case. An additional requirement relates to the feedback to the patient of the success of the call. Success is reached when the call has reached the end station, e.g. the control panel of the nurse.

A.2.3.3 Use Case 3: Nurse - Patient Interaction

Once the location of the patient and the nurse are known, new location-aware applications can be realized. Determining a nurse’s context with respect to a patient is such an example. When the nurse is located close to a patient, the patient file corresponding to the patient that is closest to the nurse could automatically open. In this way, the technology can save valuable nurse time. Besides increasing the nurse’s efficiency, the location based context can also increase the overall user friendliness and improve the usability aspects of the application. Indeed, instead of having to navigate through application screens on her PDA, the nurse can focus her attention to the patient while her PDA is detecting the patients presence and opening the correct patient file by the time the nurse reaches for her PDA to enter the information on the patient’s condition. Such usability aspects are likely to be key enabling features that can change mobile nurse-information-applications from an ICT-burden into a powerful tool that supports the nurse in her job while providing increased traceability and administrative information for the hospital quality management.

A.2.4 Classification of Use Cases

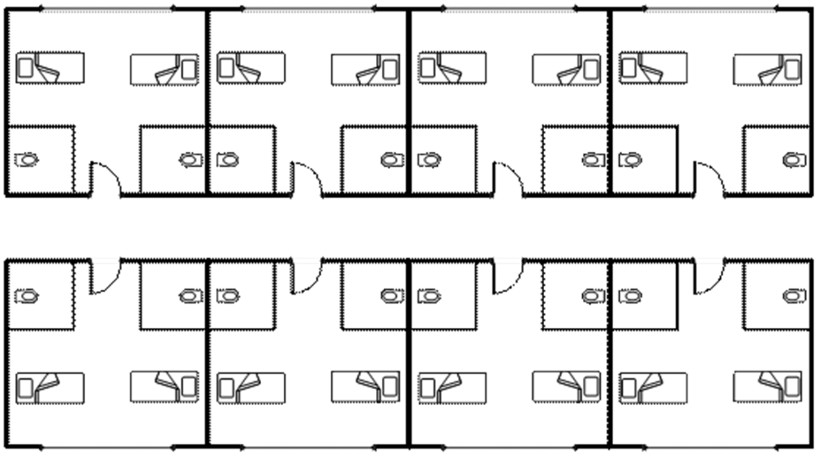
In the above section, the basic nurse call use cases have been described. However there are different use cases where the location is used to trigger an action. Note that for optimal adoption, localization is not limited to patients but also required for staff and assets, especially when localization will be used to improve the efficiency of the care giving, ensuring safety, quality and hygiene.

From a technical perspective, the use cases can be classified into three categories.

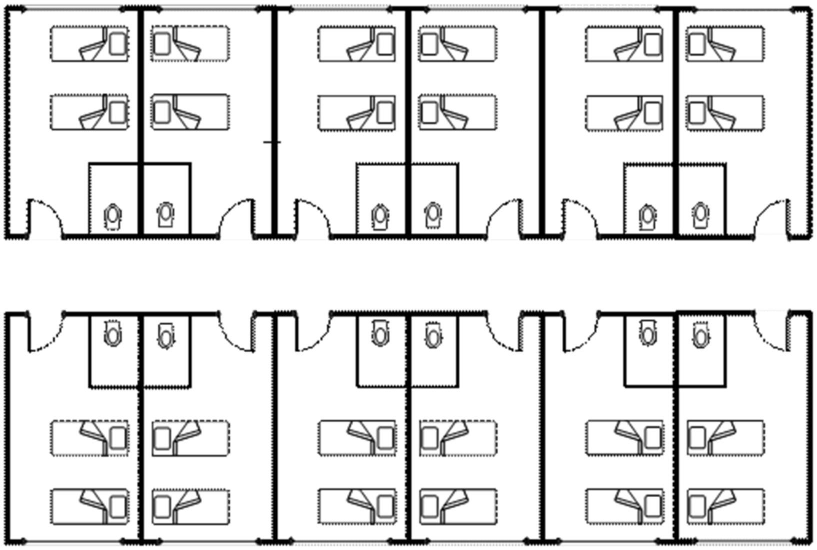
- Use cases where localization accuracy is crucial. For example, in case of an alarm call, it is required to localize the patient with at least room level accuracy. Entering the wrong room should be avoided in any circumstances. Of course, when the accuracy can be further improved to e.g. which bed, it gives an added value. Also the location of the nurse and assets (and their status) should be known with this level of accuracy to allow an improved efficiency and safety. The basic use cases as described in the previous section belong to this category.
- Use cases where the latency is critical and where the user wants to register himself to the system. Examples are access control and log-on applications. In this group of use case, it is assumed that the user initiates the action and hence the location range is limited to “close proximity”. The user should be detected inside a perimeter of 10 to 20 cm of a beacon. That beacon will then e.g. activate a lock. Typically solutions that are good in the first category fail in this category as they apply duty-cycling in order to minimize the power consumption.
- Use cases where an action automatically is triggered in the proximity of the user, without user interaction. The main use case is wandering detection where the system should detect that a patient is leaving a zone and launch an alarm. When this happens close to the exit, a small latency is required to e.g. lock a door. Compared to the previous group, the detection range should be much larger (typical HF RFID technologies are not sufficient). Again this will impact duty cycling and power consumption.

A.2.5 Description of the Environment

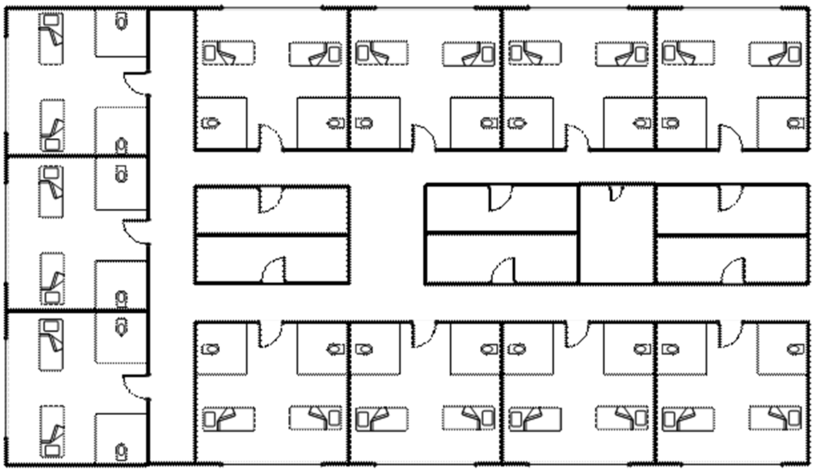
The topologies below are valid for the patient rooms in a hospital. It is important to understand that almost one third of the hospital area does not exist out of room. It can be storage rooms, public areas (cafeteria, reception), etc. This will have an impact on the localization principle as the type of rooms, the available infrastructure and the number of people in the same location differs. Dependent on the type of rooms, also the sources of interference will be different.



(a) Situation 1



(b) Situation 2



(c) Situation 3

Figure A.2: Different environment situations

All kind of structures are possible.

- The walls can be made of plaster or can be concrete walls. Note that combinations are possible and that the exact type is not always known (older buildings)
- The sizes of the room differ (dependent on building and number of beds)
- The entrance of the sanitary cell is located on a different position
- Typically long corridors are present
- Occasionally, the building has a circular or star shape
- Metal objects can be present and can be moved at any time
- The number of people present can be different (can be a lot in case of visitors)

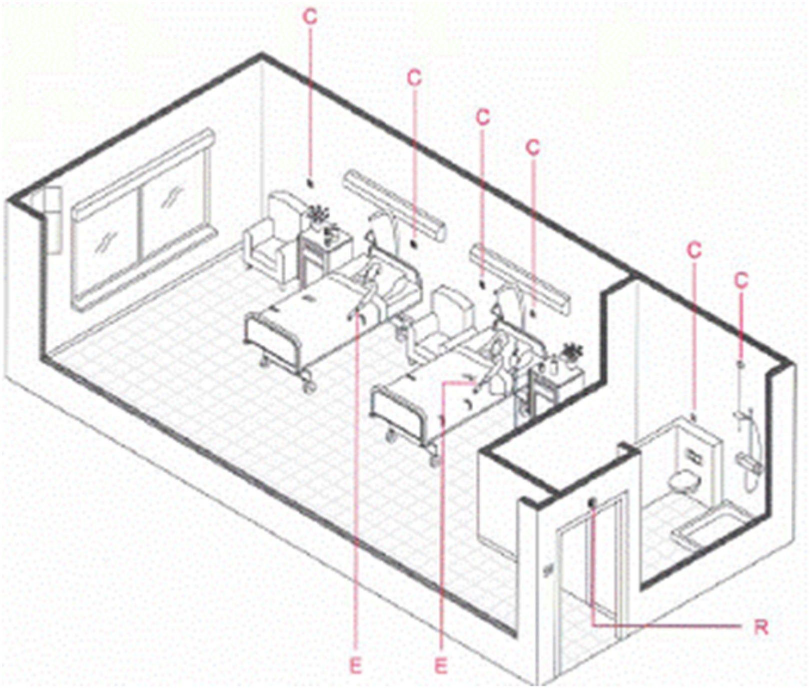


Figure A.3: Environment infrastructure

The corridor topologies shown here are limited to 2D. Obviously RF propagation will also penetrate the floor above or below but most solution offer a technique to detect the floor (e.g. passing a gate) such that the localization algorithm can be done in 2D and not in 3D. (There is one exception where the z-axis is important and that is for fall detection based on detecting that the tag is located maximum 30 cm above the floor).

A.2.6 Terminology

Wandering and access control differ in detection range and in user interaction.

- Wandering does not need an action of the person. Wandering can be defined as an “open door” environment where the system needs to take specific actions such as locking a door or launching an alarm call.

Everything goes automatic without a specific handling by the person. The range will be in the order of 1-2m (room) to 2-4m (corridor).

- Access control requires that the person makes an intension to enter a room. Access control can be considered as a “closed door” environment. The detection range is limited to about 10-20cm.

A.2.7 List of Use Cases

1. Localized nurse call
   (a) Patient makes a call, gets localized and action is triggered
   (b) Nurse makes a (e.g. anti-aggression) call, gets localized and action is triggered
2. Wandering
   (a) Patient is leaving a zone
      i. (Specific) alarm is triggered
      ii. Alarm is triggered when a patient does not return in a zone within a specific time
   (b) Patient wanders. Door should be closed when patient approaches. (patient does not do any action to enter)
3. Access control
   (a) Patient only wants access to its own room
   (b) Nurse wants to enter all rooms
   (c) Different priorities and access rules
4. Auto log-on
   (a) Nurse is automatically logged on when she is standing close to a terminal
   (b) Nurse is automatically logged on when she is standing in the room (access patient file)
5. Assets
   (a) Staff wants to request the location of an asset or a group of assets
   (b) Staff wants to know the closest e.g. wheel chair
   (c) Theft alarm
6. Track and trace
   (a) For both persons and asset, it can be interesting to follow them real time
   (b) For both persons and asset, it can be interesting to play the history e.g. in case of an infection or malfunction, one can check for whom the equipment was used

These groups of scenarios list the main healthcare use cases. Of course other related scenario scan be defined but they will typically fit into one of the above categories. Note that more advanced scenarios exist (such as work flow optimization), however they do not put new requirements on the localization system or do not define a new category. They ask more advanced reasoning on the provided location to define the correct action or retrieve interesting information from it.

A.2.8 General (Technology Independent) Requirements

- Localization accuracy dependent on the use case (see above). The minimum accuracy will be room level accuracy
- Support for wandering and access control
- Latency dependent on the use case
  - In case of a duty cycled system: how long does it maximally take before a location update is known to the system after e.g. pushing an alarm
- Acknowledgment in case of alarm
- Waterproofness
- Small and elegant form factor
  - Easy to put on for nurses and default patient
  - Removal alert or not possible to remove for critical patients
- Maximum life time
- Minimal cost
- Logging
- Easy deployment (tags and infrastructure)
  - Preferably no calibration
  - Preferably no additional cabling
- Scalability: solution works from few localized nodes up to 10.000 nodes

A.3 Application Domain 2: Underground Mining

A.3.1 Introduction

In an underground mine like El Teniente [11], a big number of miners are generally working in shifts for the exploitation of copper. In case of disaster, it is very difficult for the mine management to identify the actual person trapped, their number and exact location. Occasionally, some miners come out from the mine before completion of the scheduled shift time. In case of disaster during that period, there is no track of early adjourns of duty by such miners and the mine managements are always in doubt about how many persons are trapped. Moreover, during normal operations, workers are regularly exposed to certain conditions that can lead a whole group of people to death, e.g. the exposure to gases coming from machinery or HVAC systems or even levels of temperature and humidity that can make structures to collapse and leave people trapped inside. Therefore, the identification of the miners and the monitoring of environment conditions are vital needs for underground mine management in case of disaster as well as normal operating conditions. Furthermore, mining industry is generally capital intensive, and numbers of equipment related to production and transportation are deployed in the underground. It has been reported in many situations that the cost of maintenance at mechanized mines comes to about 35% of the operating cost of the system and it goes as high as 50-60% when both direct and indirect costs are taken into account. Sometimes it constitutes 30% of the total production cost. In today's globally competitive market scenario, efforts to reduce production cost have awaken the mining industry for automation and optimum utilization of equipment by increasing its availability and performance. Therefore, continuous monitoring of equipment location and their operation with respect to dynamic working places is necessary to make the underground mines viable, competitive and profitable.

A.3.2 Wireless Sensor Networks for Underground Mine Localization

Why it is becoming an alternative for this application?

Due to the increasing advances in wireless sensor network technologies, low power and cost effective devices are now available in the market for multiple applications. Underground mining localization is one application which has many chances of becoming a potential market for these devices.

In addition to mining, there are auxiliary activities that could be supported by these technologies:

- Health and safety: ventilation gas control, dust suppression and noise reduction
- Ground control: supporting and scaling
- Power supply and lighting
- Drainage and flood control
- Maintenance and repair of equipment

In order to provide better working conditions there are several parameters that must be measured such as CO2, water, dust measurement and also provide information of where personnel and machinery are positioned in case the mine walls collapse or for productivity purposes.

In an environment such as a copper mine there are large tunnels which length can reach several kilometers and they are wide enough to fit large vehicles that serve for production. Therefore, there are several measurement points that must be set in order to have a clear view of what these tunnels status are that require deploying a large number of sensing devices along the mine.

Current deployments imply handling monitoring devices manually due to the lack of techniques to set a large scale self-healing sensing network. When wired solutions are employed, connections require large amounts of wires and maintenance costs are very high. Indeed, scalability represents and problem in this scenario as tunnels go further and more devices are needed. Moreover, mines walls usually collapse and network operations can be seriously undermined to the point a relevant area could be lost.

This situation can be reversed by employing wireless technologies which can decrease costs drastically. However, these technologies face different problems which are related to waves' properties and the environment characteristics.

Why this technology is not fully adopted?

The mines tunnels structure itself represent an obstruction for waves propagation. The UHF wave length is far smaller than tunnels dimensions thus the transmission is similar than through a waveguide where power losses are infringed by refraction. In addition, wall roughness and uneven tunnel cross section leads to an increase of the longitudinal attenuation.

Other factors that affect network's operations are those related to the environment and people's activities. Machinery and high voltage power cables usually infringe interference in the propagation of the signal as well as dust or vapor which attenuate some frequencies.

From a practical point of view and taking into account previous technical problems these networks face, the full adoption of these technologies are also hindered by their performance in such environment. Each scenario would have to have a customized solution as not all mines have the same environmental properties and also localization accuracy is not the one required for these critical safety and production applications.

A.3.3 Use Case 1: Personnel Location and CO_2, Humidity and Temperature Measurements

Workers in this mine are usually working in shifts. All groups that enter the mine in one shift are all distributed in trains or buses to their correspondent sector. In this case, the mine operator needs to know whether the right personnel are heading the right direction or got stuck due to an incident. It also requires knowing whether the shift was completed by all workers or if safety conditions are guaranteed in their working environment.

For locating personnel, it is sufficient that the operator has identified all mobile nodes by a single identifier, location and mine sector assigned. In this case, knowing the location is the most critical aspect since mine walls could collapse and leave workers trapped. The last data collected from sensors around could be sufficient to know where personnel have suffered the accident yet it has to be accurate enough in large galleries in order to know how to react.

Regarding environments' conditions, it is necessary that the operator has information of sectors and tunnels air quality. This is a critical application that enables the detection of toxic gases in the air and to know where exactly they are originated. For this purpose, the operator has to know not only air pollution levels but also the sector where the gas has been detected and whether personnel is around in order to evacuate.

A.3.4 Use Sase 2: Machinery Location

Machinery productivity depends on a large extent on how efficiently they reach destination and how much time they are operative. Vehicles in mines are usually employed for supplies and personnel transport or mining itself. The normal operation of this equipment is traveling from one area or level of the mine to another. Independently of the application, it is necessary to know equipment location at all times and what are surrounding conditions.

For productivity reasons, it is necessary to know how much time equipment is used for its purpose. This can be measured by knowing its location at all times. In case a vehicle is being held in an area for no reason it could be reallocated for its use in another area or a failure in its functions would have been detected. Furthermore, this situation can be used to detect any outages in the service by knowing whether there has been an accident which is stopping vehicles from accessing other areas in the mine.

For safety reasons, environment conditions are measured in order to have an idea of the status of an area vehicles are passing through. In case toxic gases are measured in that same area, an alarm can be sent to the vehicle in order to guarantee operators safety.

A.3.5 Classification of Use Cases

The application thought for these technologies are indeed critical yet the environment is different from buildings. From a technical point of view there are two types of use cases:

- Use case where localization accuracy is not that critical since areas to be covered are wide enough to enclose personnel and machinery. An error of 2 to 4 meters is still acceptable.
- Use cases where network robustness is necessary. The localization algorithm has to provide its consistency through long time periods.

A.3.6 Description of Environment

The underground mining safety use case will take place in "El Teniente" mine in Rancagua, 85 km. away from Santiago de Chile. With its 2.400 km of underground galleries, it is considered the largest copper mine in the world and it currently has around 10000 employees[1]. This mine is currently operated by Codelco, a state owned company that usually does sub-contracting or leases other companies according to the required expertise. In this case, machinery and personnel are all managed by Codelco.

As seen in Figure A.4, the structure is comprised of four levels which one of them is under construction. Every level is equipped with HVAC systems and there are also several offices underground.

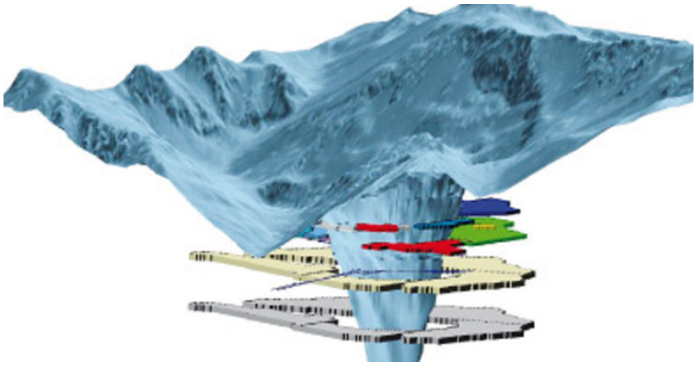


Figure A.4: El Teniente Levels

The system is intended to be deployed within one level. Each gallery will be equipped with different network nodes measuring presence, humidity, temperature and CO_2 levels. The physical topology will follow the mine structure situating data concentrators or routers in the joints of several tunnels. These tunnels can vary in size but the average can be of around $4x4m$ and interferences are quite high due to walls thickness and roughness.

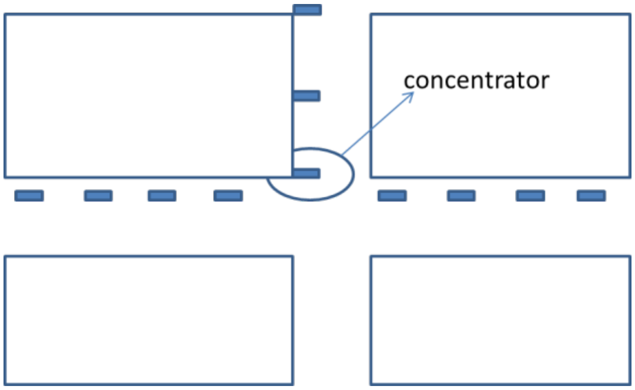


Figure A.5: Concentrators

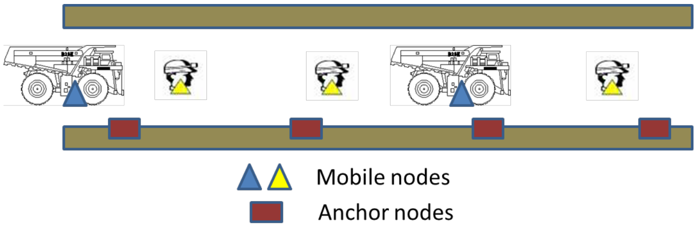


Figure A.6: Topology of the nodes

---

[1] `http://www.codelco.com/prontus_codelco/site/edic/base/port/el_teniente.html`

The actual scenario where the solution will be deployed can be seen in the following pictures. The entrance will be the gate for access control and anchor nodes in tunnels will provide personnel and machinery localization.

The tests will be applied to a limited group of workers and machinery. Personnel tests in particular will be applied to a group which will be carried to a specific area inside vehicles (bus or train).

Figure A.7: Main entrance

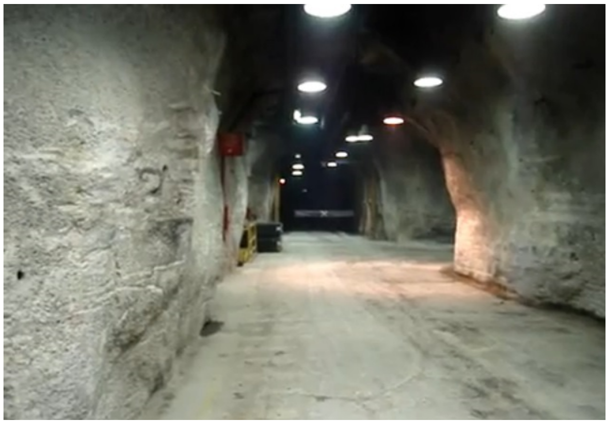


Figure A.8: Average tunnel

A.3.7 List of Use Cases

1. Personnel tracking: Personnel are tracked and traced from the entrance to their destination
2. Machinery tracking: Machinery is tracked and traced from the entrance to their destination
3. Air quality monitoring and personnel location: An alarm is triggered when certain levels of toxic gases are reached. Personnel are evacuated and traced.
4. Air quality monitoring and machinery location: An alarm is triggered when certain levels of toxic gases are reached and machinery is approaching that area.

References

[1] “IEEE Standard Glossary of Software Engineering Terminology," IEEE Std 610.12-1990 , 10.1109/IEEESTD.1990.101064

[2] The CREW project, `http://www.crew-project.eu/`

[3] ISO/IEC JTC 1/SC 31 Automatic identification and data capture techniques - `http://www.iso.org/iso/iso_technical_committee.html?commid=45332`

[4] The ISO Standard - ISO/IEC CD 18305 - Information technology - Real time locating systems - Test and evaluation of localization and tracking systems - `http://www.iso.org/iso/catalogue_detail.htm?csnumber=62090`

[5] Taylor, Barry N. - Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results (DIANE Publishing, 2009.) [ONLINE] `http://books.google.be/books?id=xgJeJzsbhoAC`

[6] The EVARILOS project, “D2.2. Report on Experiments Without Interference”

[7] The EVARILOS project, “D2.3: Report on Experiments With Controlled Interference”

[8] The EVARILOS project, “D2.5: EVARILOS Benchmarking Suite”

[9] The EVARILOS project, “D4.1: Report on the definition and setup of the validation scenarios”

[10] The EVARILOS project, “D4.3: Report on the Results of the Open Challenge”

[11] Mines in El Teniente, `http://www.mining-technology.com/projects/el-teniente-new-mine-level-project/`

[12] E. Agu, “Advanced sampling”, `http://web.cs.wpi.edu/~emmanuel/courses/cs563/S10/talks/wk3_emmanuel_poisson_disk.pdf`