

# Evaluating Indoor Positioning Errors

Teemu Pulkkinen and Johannes Verwijnen

Ekahau Oy

Hiilikatu 3, 00180 Helsinki, Finland

firstname.lastname@ekahau.com

**Abstract**—With the advent of the indoor positioning field there has been a plethora of different approaches to locating people and assets in indoor environments, each with their own strengths and weaknesses. Despite this, a lot of the metrics used for optimization may contain hidden biases (unintended or neglected) or end up measuring aspects of the solution which conflict with the intended use case. In this position paper we hope to initiate a discussion on standardized baselines and metrics, and suggest some normalizing measures to alleviate bias and further a standardization of error metrics.

## I. INTRODUCTION

In recent years, the indoor positioning field has seen remarkable progress. Early advances in positioning accuracy and robustness have to some extent given way to focusing on deployment costs and ease-of-use, but we argue that some of the fundamental aspects of indoor positioning are still lacking. We'll focus mainly on two aspects:

1) The positioning accuracy and precision [1], as used in research, do not always fall in line with the perceived usefulness of the positioning solution in the eyes of the end user. Though a strict Euclidean error might in many cases be the mathematically optimal measure to minimize, it can, if unchecked, lead to strange and unwanted behavior.

2) Though several advances have been made in improving the accuracy of positioning systems, it is hard to compare approaches directly because of inconsistent deployments. That is, the scale of the environments differ and (in the case of WiFi positioning in particular), the density of the positioning architecture can have a large influence on the results.

In this position paper we hope to bring up some (normalized) distance metrics and baselines to alleviate some of the perceived issues in future endeavors, or spark discussion on the way algorithms are currently evaluated if nothing else.

We have divided our topic into the concepts of "one-off positioning", i.e. errors incurred when discrete estimates are compared to known ground truth locations, and "tracking"; the continuous positioning of a user in terms of a sequence of estimates.

## II. ONE-OFF POSITIONING

Though a positioning approach might not directly apply a map-matching scheme in its localization phase, an indirect bounding of the solution space usually occurs through the selection of reference points (*RP*). In other words, the physical space itself limits the scale of the errors that can be achieved,

and it would thus be useful to regard the size of the environment in question when presenting localization errors.

An error evaluation that relies solely on the Euclidean distance between the ground truth position and the estimate does not consider where the estimate is located with respect to ground truth; it only sees the shortest path.

In the following, we present some solutions for normalizing the positioning error, as well as some naive baselines to rule out obvious biases in positioning solutions. In the provided examples, the location information used for calculations has been generated manually to highlight the issues discussed.

### A. Normalization

In the following we suggest some ways to normalize errors to make them comparable across deployments as well as alternate error metrics which adhere more closely to the intended use case.

Though these approaches are presented separately, one could also use a (possibly weighted) combination depending on the use case.

1) **Environment-normalized error:** Errors could be divided by the diagonal of the bounding box of the environment, or the largest possible error that could be made in estimation (the longest path from one accessible location to another). This normalization has the benefit of producing an error in the range [0,1], which is then directly comparable to other environments. A caveat here is that very large environments might receive a good score simply due to scale.

See Fig. 1 and Table I for a simulated example. By simply adjusting the scale (10 and 100 pixels per meter, respectively) of the floorplan in question we achieve wildly different interpretations of the positioning accuracy. Conversely, scaling the errors by the environment diagonal (**Diag.**,  $\approx 1496$  pixels in this example) produces a normalized score that is comparable across environments.

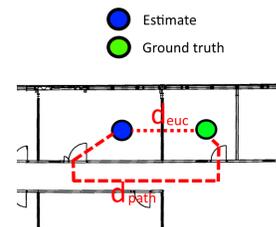


Fig. 2. Euclidean distance does not consider environment constraints.

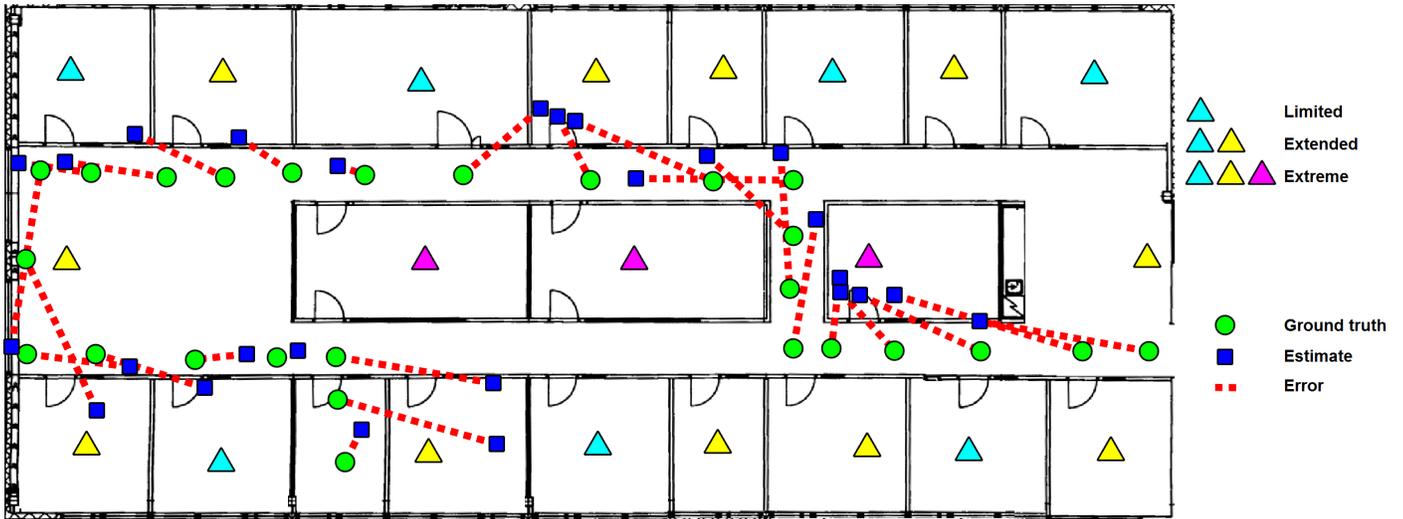


Fig. 1. Layout of simulated example.

2) **Shortest path error:** In some cases, an error estimate based on the Euclidean distance does not reflect the perceived error from the point of view of a positioning system user. An indoor space may be segregated by walls in such a way that moving between adjacent rooms incurs a considerably higher cost than the Euclidean distance suggests. This path distance also provides an intuitive interpretation of the error for estimates that localize to the wrong floor (or building). "Floor jumps" would essentially be penalized by the shortest distance involving traversal through staircases or elevators. See Fig. 2 for a visual representation.

3) **Zone/cell accuracy:** Based on our experience, indoor positioning system end users are sometimes more interested in zone accuracy than absolute positions. Users may want to be very certain about whether a person or asset is inside a certain room or not. In this use case, the actual position within that room has less importance. This error could be represented through pure classification error (binary per estimate, percentage as aggregate) when the entire space has been discretized, or by the distance to (or between) the middle of the zone when not.

### B. Baselines

In addition to the errors described above, we also suggest certain standard (naïve) baselines for comparison, to rule out biases caused by the environment setup.

1) **Nearest AP error:** In their original RADAR paper [2], Bahl & Padmanabhan used a "strongest AP" positioning solution (citing [3] as their inspiration) as a baseline comparison to their k-nearest neighbors approach to WiFi fingerprinting. Though this approach seems naive from the perspective of more modern solutions, it nevertheless functions well as a first sanity check for any promising (WiFi) indoor localization solution. In an environment with a dense enough access point infrastructure the mean accuracy can reach impressive values, with the added bonus of unrivaled robustness. In other words, comparing estimates to the location of the strongest APs intends to answer whether the current estimate is better than if we had positioned according to the strongest AP. One could

	10:1	100:1	Diag.		Limited	Extended	Extreme
5%	3.02	0.30	0.02	5%	11.16	9.16	7.86
50%	11.97	1.20	0.09	50%	16.96	12.61	11.14
95%	20.05	2.00	0.14	95%	24.17	23.07	13.50
avg	12.21	1.22	0.09	avg	16.66	14.10	10.96
				better	27%	42%	58%

TABLE I. SCALE COMPARISON.

TABLE II. NEAREST AP ERROR (10:1).

even normalize the error by the distance from the strongest AP to the ground truth location, to determine a score representing how much better a specific estimate is than the naïve one.

Fig. 1 and Table II display a simulated example of this comparison. In this example, three sets of access points with increasing density were evenly placed on the floorplan. Since this simulated example used no real data, we made a simplifying assumption that the access point closest to the ground truth exhibited the strongest signal. Positioning to the closest access point achieved the results in Table II (10:1 scale). What this example intends to show is that in a dense enough deployment (**extreme**) simply locating to the closest access point could provide a good estimate, and thus a reasonable baseline.

2) **Random/center error:** In implementations depending on discrete locations, estimates could be compared to a random approach where the estimated location is random. This baseline would serve to highlight how much better the presented approach is compared to pure chance. This technique was also used in RADAR. Alternatively, a baseline that always locates to the center point (or mass center point of the polygon) of the environment would additionally address issues of scale.

### III. TRACKING

Measuring one-off positioning accuracy has been the norm for indoor positioning evaluations. However, many indoor positioning systems are used for tracking users or assets. Thus the positioning system may use historical position estimates and knowledge of possible movement speeds and obstacles to limit the set of possible locations.

The metrics described earlier focused mainly on the pure error an estimate produces with respect to a known ground

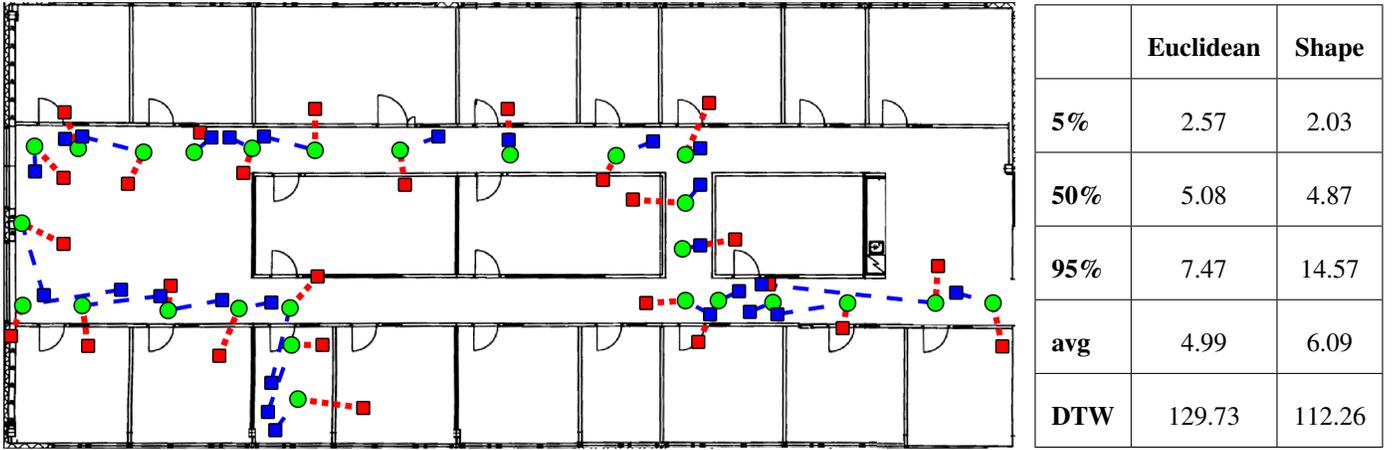


Fig. 3. Simulated tracking example. Red squares (dotted error) represent a Euclidean optimization; blue squares (dashed error) a shape-preserving focus.

truth location. For sequential estimates, or trajectories, we would in addition be interested in the overall "shape" of the route, i.e. how well the position adheres, over time, to the actual route the user takes. A positioning algorithm optimizing these scores could thus produce a more "natural" tracking behavior, and reduce the erratic aspects endemic to certain classes of positioning systems.

1) **Continuous environment-normalized error:** A sequence of estimates could form a cohesive polygon, either through merging error circles sequentially, or determining vertices based on the distance from the ground truth location to the position estimate. By comparing the area of the resulting polygon to the area of the target environment, with respect to the ground truth polygon (of width  $\epsilon$ , i.e. the estimated uncertainty of ground truth measurements) we could determine a score for the trajectory that takes into account the scale of the environment.

2) **Movement vector error:** Tracking could be considered a sequence of motion vectors between estimates. Standard geometric tools such as dot products and cosine angles could be used to penalize tracking that changes direction frequently but which, in terms of Euclidean distance, would otherwise have the same cost as a path that stays at a fixed distance from the ground truth path. Optimizing this score would serve to improve robustness.

3) **Dynamic time warping:** Treating the trajectory of a user as a time series allows the use of traditional time series algorithms such as dynamic time warping (DTW). The essential idea is often to determine the amount of change one trajectory would have to undergo to transform into the other. DTW, for instance, is essentially edit distance with a dynamic cost, which would here be defined as the distance between the estimate and ground truth. This score would to some extent be robust against lag-based errors, since DTW is designed to handle variances in speed.

In Fig. 3, and the adjacent table, a simulated example is presented where two hypothetical positioning techniques are compared in terms of Euclidean distance and DTW. Although the Euclidean-optimized technique clearly performs better in the average and worst cases, it has a worse DTW score than its shape-preserving counterpart. Using this score could be crucial

in access control scenarios where the exact position estimate is less important than detecting whether certain areas have been entered and/or exited.

4) **Trajectory clustering metrics:** Approaches such as TRACLUS [4] have been successful in clustering trajectories, mainly through good use of trajectory distance metrics. In TRACLUS, the authors define a weighted sum of geometric distances between trajectory segments (lines) as the distance between two trajectories. This score could also be used for scoring a path estimate, with different weights used to optimize the positioning algorithm for specific environments.

#### IV. FUTURE WORK

In the future, we hope to compile a survey on the most popular indoor positioning techniques, and compare them using some of the techniques presented here. We also intend to develop some metrics of our own, hopefully establishing some standardized technique for comparing position algorithm fitness over environments.

In addition, we would like to discuss the perceived utility of position estimates from the perspective of end users in contrast with the focus of the machine learning optimizations behind the actual algorithms.

#### ACKNOWLEDGMENTS

This work was supported by the Finnish Funding Agency for Technology and Innovation (project D2I).

#### REFERENCES

- [1] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [2] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System." in *Proceedings of the 19th Conference on Computer Communications (INFOCOM)*, vol. 2. IEEE Computer Society, 2000, pp. 775–784.
- [3] T. D. Hodes, R. H. Katz, E. Servan-Schreiber, and L. Rowe, "Composable ad hoc mobile services for universal interaction," in *MobiCom*, ser. Lecture Notes in Computer Science, 1997, pp. 1–12.
- [4] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD)*. ACM, 2007, pp. 593–604.