# Data Information Fusion from Multiple Access Points for WiFi-based Self-localization

Renato Miyagusuku, Atsushi Yamashita and Hajime Asama[1]

***Abstract*— In this work we propose a novel approach for fusing information from multiple access points in order to enhance WiFi-based self-localization. A common approach for designing WiFi-based localization systems is to learn location-to-signal strength mappings for each access point in an environment. Each mapping is then used to compute the likelihood of the robot's location conditioned on sensed signal strength data, yielding as many likelihood functions as mappings are available. Office buildings typically have from several tens to a few hundreds of access points, making it essential to properly combine all available likelihoods into a single, coherent, joint likelihood that yields precise likelihoods, yet is not overconfident. While most research has focused on techniques for learning these mappings and improving data acquisition; research on techniques to adequately fuse them has been neglected. Our approach for data information fusion is based on information theory and yields considerably better joint distributions than previous approaches. Furthermore, through extensive testing, we show that these joint likelihoods considerably increase the system's localization performance.**

## I. Introduction

The ubiquity of WiFi networks in indoor environments makes their use appealing for robot localization systems. A common technique for designing these systems is to learn a location-to-signal strength mapping for each access point in the environment directly from samples obtained at known locations. This technique is known as fingerprinting, scene analysis or profiling, and has been widely preferred by the community due to its higher localization accuracies in practice [1]. For each access point, the likelihood of the robot's location conditioned on sensed signal strength from the said access point can be computed using its corresponding learned mapping. Therefore, as many likelihoods as access points are obtained. With typical office buildings often having several tens to few hundreds of access points, all these individual likelihood functions must be combined into a single, coherent, joint likelihood. This joint likelihood can then be used as the perceptual model of any localization algorithm. Figure 1 shows an overview of this approach.

Previous work in the area has been focused on proposing new fingerprinting techniques, such as linear interpolation in graphs [2], vector field maps [3] and Gaussian Processes (GPs) [4], [5]. Or extending previous approaches to increase their precision [6], robustness [7], or address the SLAM problem [8], [9]. Another active area of research centers around data and training datasets acquisition, for these techniques' performance heavily relies on them. Previous work in this area includes filtering data samples and eliminating redundant access points [10], continuous incorporation of data samples into the training datasets and re-training of the models [11], the addition of training samples based on the expected reduction of localization uncertainty [12], among others.

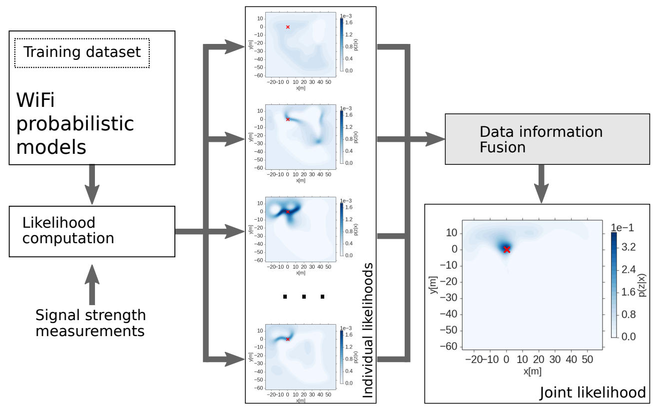


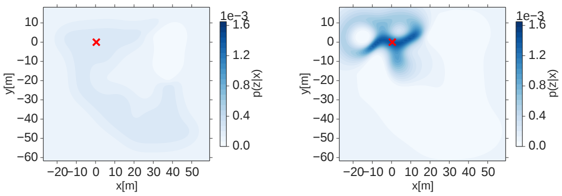

Fig. 1. WiFi fingerprinting approaches learn location-to-signal strength mappings for each access point available from a training dataset. Using these mappings, the likelihood of locations is computed given new sensor measurements. Finally, all likelihood functions are combined into a joint likelihood distribution that can then be used by any probabilistic localization algorithm. In this work we focus on data information fusion algorithms which enable the computation of efficient joint likelihoods.

Although of vital importance, seldom work has been done related to methods to combine all individual likelihoods. To the best of our knowledge, only two approaches have been used for computing this joint likelihood distribution. The first assumes mutual independence between the access points' signals given locations, and therefore, computes joint likelihoods as the product of all individual likelihoods or the log of its product [4]. The second also computes the joint likelihoods as the product of individual ones but elevates each one to the power of a "smoothing" coefficient. When this coefficient is the reciprocal of the number of access points, as suggested in [5], the model computes joint likelihoods as the geometric mean of the individual likelihoods. Interestingly, this is equivalent to a Product of Experts model (PoE) [13]. Under this approach, each mapping is considered as an "expert" which makes inferences regarding the same situation, for our particular case, the robot's location.

Both mentioned approaches weight all models equally; however, we consider that these approaches are suboptimal

This work was funded by Tough Robotics Challenge, ImPACT Program of the Council for Science, Technology, and Innovation (Cabinet Office, Government of Japan).

[1]R. Miyagusuku, A. Yamashita and H. Asama are with the department of Precision Engineering, Graduate School of Engineering, The University of Tokyo, Japan. {miyagusuku, yamashita, asama}@robot.t.u-tokyo.ac.jp




(a) Access Point 0 (b) Access Point 160

Fig. 2. Likelihood distributions generated by two different models, for the same test point, with darker shades representing areas of higher probability. Ideal distributions should concentrate most of its probability close to the robot's true location (red X). Difference in quality of the predictions is noticeable, with (a) providing little additional information, as it is almost a uninformative distribution; and (b) much more, as it narrows down locations to the upper left corner of the locations space.

as some models provide more information than others and should be given more importance. To illustrate this, Fig. 2 shows an example of the likelihoods generated by different access points in an office building. As it can be observed, the likelihood shown in Fig. 2a provides little information regarding the robot's locations as it computes very similar likelihood values for every possible location. Differently, the likelihood is shown in Fig. 2b provides a fair amount of information, as it assigns higher likelihood values to locations in the top left corner of the locations space.

To incorporate the idea that some models provide better likelihoods than others, we use an extension of the PoE framework, named general Product of Experts (gPoE) [14]. gPoE weights each expert's contribution differently by computing the joint distribution as a weighted geometric mean. This allows us to regulate the contribution each individual likelihood has by modifying these weights. Weighted geometric means have also been used to extend another fusion framework known as the Covariance Intersection algorithm [15]. Furthermore, under this approach, it has been proven that weighted geometric means are conservative and effective in the context of data fusion of dependent sources [16], which is our case.

Finding an adequate approach for computing these weights is the most important design choice when employing any of these frameworks. Works like [14] and [17] have suggested the use of measures related to the generated likelihoods' entropy for weights computation. Instead, we chose to employ the difference in the entropy between the generated posteriors and an uninformative prior. This measure can be interpreted as the amount of information gained by making a specific measurement. The posteriors' entropy is employed instead of that of the likelihoods because the entropy of all our likelihood functions is the same. This is a direct result of the way we model access points but is rooted in the fact that all our sensors are similar and work in the same environment.

The remaining of this paper is organized as follows. In Sec. 2 we derive the different data information approaches that have been employed for WiFi data information fusion, as well as gPoE, based on the dependence assumptions placed on the sensor measurements from each access point. In Sec. 3 we focus on weights computation for gPoE models, starting by discussing another tentative approach, to then focus on our proposed measure and both its discrete and continuous implementations. In Sec. 4 we evaluate the effectiveness of our approach when used together with a Monte Carlo Localization (MCL) algorithm [18], as well as how it fares with previous approaches. Finally, Sec. 5, discusses the obtained results and concludes this paper.

## II. Models for data information fusion

In this work, we focus on data information fusion from several probabilistic WiFi models. For this, we assume a mapping has already been learned for each access point, and do not address the particular mechanisms used for learning these mappings. The mappings used in this work have been learned using Gaussian Processes, which given some training data, generalize these into a continuous function where each point is considered to have a normal distribution. Hence, our mappings need only to generate mean and variance predictions of the access point's signal strength as measured for any arbitrary location $\mathbf{x}$. In this work, locations $\mathbf{x}$ are x-y Cartesian coordinates, as we assume the antennas to have fairly homogeneous radiation patterns, making it unnecessary to consider the relative angle between them. Although this is unlikely, the models obtained under this assumption hold well in practice [6].

As every point in the mapping is assumed to have a normal distribution, the probability that the signal strength measurement $z$ was observed at an arbitrary location $\mathbf{x}_*$, is computed as,

$$p(z|\mathbf{x}_*) = \frac{1}{\mathrm{sd}[z_*]}\varphi\left(\frac{\mathbb{E}[z_*]-z}{\mathrm{sd}[z_*]}\right). \tag{1}$$

with $\mathbb{E}[z_*]$ and $\mathrm{sd}[\mathbf{z}_*]$ the predicted mean and standard deviation generated by the model at location $\mathbf{x}_*$, and $\varphi(\cdot)$ the standard normal distribution function. However, any other probabilistic model can be used instead, to compute these likelihoods. For a detailed description of the employed approach, readers are referred to [6].

As we are interested in the posterior probability of locations $\mathbf{x}_*$ conditioned on a WiFi measurement $z$, i.e., $p(\mathbf{x}|z)$. We use Bayes Rule to compute it from the likelihood, as,

$$p(\mathbf{x}|z) \propto p(z|\mathbf{x})p(\mathbf{x}). \tag{2}$$

Given $m$ access points, and hence $m$ WiFi models, our problem consists on finding a suitable joint likelihood function $p(z_1, z_2, \ldots, z_m|\mathbf{x})$, which allows proper computation of posterior distributions.

Using the general product rule of probabilities we can rewrite $p(z_1, z_2, \ldots, z_m|\mathbf{x})$ as,

$$\begin{aligned} p(z_1, z_2, \ldots, z_m|\mathbf{x}) = \\ p(z_1|\mathbf{x})p(z_2|z_1,\mathbf{x})\cdots p(z_m|z_1, z_2, \ldots, z_{m-1}, \mathbf{x}) \end{aligned} \tag{3}$$

The previous two approaches used for fusing WiFi likelihoods can be derived by making different assumptions regarding the conditional independence of signal strength measurements $z_i$.

## A. Product of likelihoods model

If we assume every two models with measurements $z_p$ and $z_q$ to be conditionally independent given a location $\mathbf{x}$, we have

$$p(z_p|z_q, \mathbf{x}) = p(z_p|\mathbf{x}) \qquad \forall p \neq q, \tag{4}$$

which yields the product of likelihood models,

$$p(z_1, z_2, \ldots, z_m|\mathbf{x}) = \prod_{i=1}^{m} p(z_i|\mathbf{x}). \tag{5}$$

Assuming all individual likelihoods to be conditionally independent of each other given the location, is a valid assumption; however, the posterior distributions generated from this approach often yield overconfident estimations. This is a common issue with Bayesian updating when several models are sequentially used for updating posterior beliefs. Even if the individual likelihoods are not confident in their predictions, the updated posterior computed will yield an extremely peaked distribution nonetheless. This is not an issue if the estimated location is computed directly as the location which maximizes the posterior distribution, as done in [4], as the maximum value will be the same regardless of how smooth or peaked the distribution is (although in this case, it is more common to use the log of the posterior as it is smoother and easier to search). However, these overconfident predictions are not adequate inputs for localization algorithms based on the Bayes Filter, such as the Monte Carlo Localization algorithm, as these localization algorithms use the whole posterior distribution, therefore require adequate variances.

## B. Product of Experts model

As by the general product rule of probabilities we can rewrite $p(z_p, z_q|\mathbf{x})$ as either $p(z_p|\mathbf{x})p(z_q|z_p, \mathbf{x})$ or $p(z_q|\mathbf{x})p(z_p|z_q, \mathbf{x})$, by multiplying both expressions we have that,

$$p(z_p, z_q|\mathbf{x})^2 = p(z_p|\mathbf{x})p(z_q|\mathbf{x})p(z_p|z_q, \mathbf{x})p(z_q|z_p, \mathbf{x}). \tag{6}$$

If the models are assumed to be completely dependent, $p(z_i|z_j, \mathbf{x})$ would be fully known, and the previous expression would be equivalent to

$$p(z_p, z_q|\mathbf{x}) = p(z_p|\mathbf{x})^{\frac{1}{2}} p(z_q|\mathbf{x})^{\frac{1}{2}}. \tag{7}$$

Applying the same assumptions for $m$ models, we obtain

$$p(z_1, z_2, \ldots, z_m|\mathbf{x}) = \prod_{j=1}^{m} p(z_j|\mathbf{x})^{\frac{1}{m}}, \tag{8}$$

which is equivalent to the product of experts model.

This approach solves the overconfidence issues of the previous one by smoothing the joint likelihood; however, this total dependence can generate underconfident predictions. However, in practice, when using Bayesian filters, these underconfident predictions yield much better results than the overly confident ones generated by the product of likelihoods model.

## C. General Product of Experts model

The two previous approaches weight all individual likelihoods equally. As we expressed in the introduction, we consider that this should not be the case, as for a given location certain WiFi models provide more information than others. For our approach we use the same complete dependence assumption. By arbitrarily elevating $p(z_p|\mathbf{x})p(z_q|z_p, \mathbf{x})$ and $p(z_q|\mathbf{x})p(z_p|z_q, \mathbf{x})$ to the powers of $\lambda_p$ and $\lambda_q$ respectively, and multiplying them, we can demonstrate that the following is also a valid equation for computing its joint likelihood,

$$p(z_p, z_q|\mathbf{x})^{\lambda_p+\lambda_q} = p(z_p|\mathbf{x})^{\lambda_p} p(z_q|\mathbf{x})^{\lambda_q}, \tag{9}$$

$$p(z_p, z_q|\mathbf{x}) = p(z_p|\mathbf{x})^{\frac{\lambda_p}{\lambda_p+\lambda_q}} p(z_q|\mathbf{x})^{\frac{\lambda_q}{\lambda_p+\lambda_q}}. \tag{10}$$

Then, for $m$ models,

$$p(z_1, z_2, \cdots, z_m|\mathbf{x}) = \left( \prod_{i=j}^{m} p(z_j|\mathbf{x}_*)^{\lambda_j} \right)^{1/\sum_{j=1}^{m} \lambda_j}. \tag{11}$$

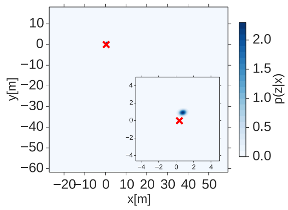
This results in an equivalent equation as the one used by the general Product of Experts framework, which computes joint distributions as weighted geometric means of the individual likelihoods. As this approach assigns different weights to each expert when aggregating their outputs, if weighted coefficients are properly calculated, it is possible to favor experts which contribute more information; generating more confident posteriors. The issue then becomes, how to compute these weights; which will be addressed in the next section.

Figure 3 shows some examples of posterior distribution obtained from an experiment performed in an indoor environment, using the aforementioned approaches for computing joint likelihoods, and assuming an uninformative prior distribution over locations. As it can be observed, when using the general product of experts approach more adequate posteriors can be obtained.

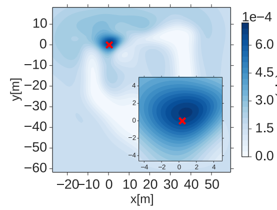
# III. Approaches for computing weights

In this section, we discuss different approaches for computing adequate weights $\lambda$ for the gPoE model.

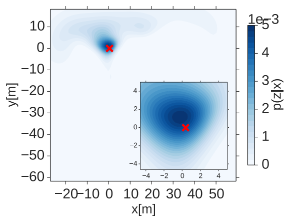
The final goal of our system is to fuse the information provided by our WiFi models (in the order of several tens to few hundreds of models), into a coherent joint likelihood which can be used to provide accurate and reliable posterior distributions of the robot's location. Therefore, although we wish to compute the joint distribution of measurements given locations $p(\mathbf{z}|\mathbf{x})$, we are truly interested in the model's posteriors, $p(\mathbf{x}|\mathbf{z})$. If some models were inherently better than others (i.e., consistently provide more information), it would be possible to use a constant weight for each model. However, for our WiFi models, we have noticed this not to be the case. The amount of information each model provides varies depending on its observed signal strength information; therefore, rather than a constant weight, we propose to compute weights by learning a mapping over the signal strength space, i.e., $\lambda = f(\mathbf{z})$.




(a) Product of likelihoods




(b) Product of experts




(c) Our approach

Fig. 3. Joint likelihoods generated using different approaches. Although ideal likelihoods should concentrate most of its probabilities around the robot's true location (red X); extremely peaked distributions as the one shown in (a) are not desirable as these are overconfident estimations.

This mapping could be learned directly from the same training data employed to learn each individual WiFi mapping. However, this function has a much higher dimensionality. While WiFi mappings are $\mathbb{R}^2 \rightarrow \mathbb{R}$ (x-y locations to signal strength), this weight mapping is $\mathbb{R}^m \rightarrow \mathbb{R}^m$ (signal strength of the $m$ WiFi mappings to $m$ relative weights that need to be considered jointly given our dependence assumption). Therefore, the required amount of training data necessary to learn a robust mapping is much higher. To avoid this issue, we first turn our attention to methods that require no training datasets.

A measure that can be employed to learn $\lambda$ without using labeled data is uncertainty. In general, the uncertainty of a function can be understood as a measure of the dispersion of samples obtained from it. If samples are consistent, it is said that the function has low uncertainty. On the contrary, if samples are widely dispersed, it is said that they have high uncertainty. Under this definition, the distribution with the highest uncertainty is the uniform distribution, as its predictions are equally dispersed around its whole output domain; while the one with the lowest uncertainty would be one where the location is fully known (perfect information), as the same prediction would always be sampled.

The variance of a function is a common metric used for the estimation of its uncertainty. Variances are good estimators of uncertainty only for unimodal functions and are sufficient only if such function has a Gaussian distribution. Even though Gaussian Processes assume each point in the mapping has a Gaussian distribution, i.e., $p(z|\mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$, there is no such assumption regarding its posterior $p(\mathbf{x}|z)$. Joint posterior distributions generated by WiFi models are mostly unimodal; however, individual posteriors distributions generated by each access points are not. Making not only variance a poor estimator of uncertainty, but also making the likelihood of predictions at the ground truth and cross entropies poor estimators of performance. To illustrate this, Fig. 4 shows several examples of individual posteriors generated by the WiFi models presented in [6]. As it can be observed, although the posteriors shown in Figs. 4c and 4e have high uncertainty (they are almost uniform distributions), their variance is lower than that of posteriors in Figs. 4d and 4f. This occurs as variance cannot properly quantify the uncertainty of the posterior in Fig. 4c because this distribution is multimodal and the one in Fig. 4e although unimodal, it is not Gaussian, which makes its variance insufficient for determining the uncertainty of its predictions.

A metric which remains reliable regardless of the distribution's modality is entropy. Entropy is a central metric employed in information theory, which, similarly to variance, quantifies the amount of uncertainty of a random variable. Specifically, Shannon entropy is employed in this work. Shannon entropy is defined as the expected amount of information encoded in a discrete random variable. For a variable $\mathcal{X}$ with possible discrete values $\{x_i\}^{i=1:k}$, and corresponding point mass probabilities $\{p(x_i)\}^{i=1:k}$, its Shanon entropy $H(\mathcal{X})$ measured in nats, is computed as,

$$H(\mathcal{X}) = -\sum_{x_i \in \mathcal{X}} p(x_i) \ln(p(x_i)). \tag{12}$$

The entropy of $\mathcal{X}$ takes its minimum value for a distribution with all its probability concentrated in a single element $x_i$; which results in $H(\mathcal{X}) = 0$ (perfect information). And, it takes its maximum value for a uniform distribution; which results in an entropy $H(\mathcal{X}) = \ln(k)$.

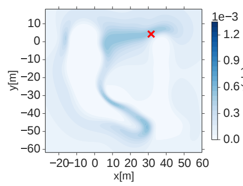
It is also possible to measure the entropy of a random variable conditioned to a measurement. For example, after the measurement $z_{obs}$ has been acquired, the resulting entropy of $\mathcal{X}$ becomes,

$$H(\mathcal{X}|z_{obs}) = -\sum_{x_i \in \mathcal{X}} p(x_i|z_{obs}) \ln(p(x_i|z_{obs})); \tag{13}$$

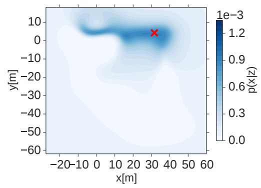
which, contrary to variance, adequately quantifies the distributions uncertainty of the posteriors shown in Fig. 4. Conditional entropy for posteriors in Figs. 4c and 4e is larger than in all other cases, as expected from our definition of uncertainty.

### A. Minimum Entropy (gPoE minH)

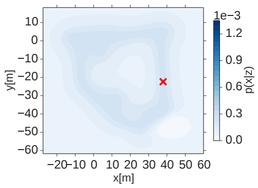
Using this more reliable measure of uncertainty, it is possible to compute gPoE weights, so that the resulting joint likelihood has the minimum possible uncertainty, i.e., minimum entropy. However, this is not recommendable, as uncertainty is related to the precision of the distributions, not their accuracy. Therefore, even if a distribution has low uncertainty, it could still produce samples far from the true one. Models that generate such distributions, under the product of experts formulations, are denominated "bad experts". With both the PoE and gPoE models being considerably sensitive to them due to their aggregation method, the product function. As experts are multiplied, each expert has "veto" power. Therefore, it is sufficient for a single bad expert to output low-value probabilities for the joint prediction to output a



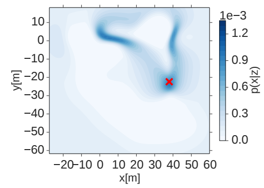

(a) Var:588 $H(\mathbf{x}|z)$:5.75
Test Point 20, AP 170



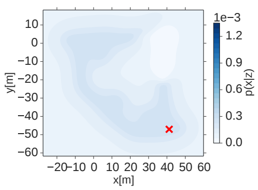

(b) **Var:480** $H(\mathbf{x}|z)$**:5.55**
Test Point 20, AP 120



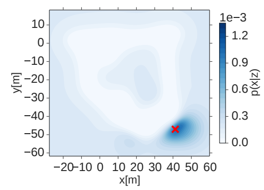

(c) **Var:511** $H(\mathbf{x}|z)$:5.93
Test Point 40, AP 080




(d) Var:548 $H(\mathbf{x}|z)$**:5.63**
Test Point 40, AP 090




(e) **Var:529** $H(\mathbf{x}|z)$:5.92
Test Point 60, AP 000



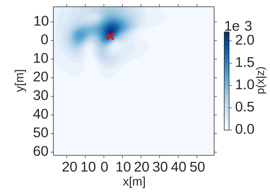

(f) Var:697 $H(\mathbf{x}|z)$**:5.73**
Test Point 60, AP 080

Fig. 4. Example of posterior distributions conditioned on a single access point (AP). Two figures are shown per test point. Those on the left (a,c,e) having less uncertainty than those on the right (b,d,f). All posteriors show their computed variance and conditional entropy (the model with lower uncertainty is highlighted in bold). As observed, conditional entropy reliably assigns a lower uncertainty to models on the right, while variance does not.

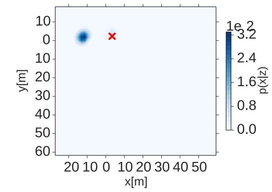
low value. If experts are trustworthy, this is a very desirable characteristic, as it allows for joint likelihoods to produce highly confident predictions. However, in the presence of bad experts, it compromises the robustness of the approach. Figure 5 shows an example of this occurring in practice; where the joint distribution of minimum entropy under the gPoE model (Fig. 5b) places over 80% of its weight on a single model, which was a bad expert. When compared to the resulting distribution under the standard PoE approach (Fig. 5a), the issues with this approach become evident.

## B. Information Gain (gPoE deltaH)

The gPoE model can overcome the effect of bad experts by assigning low weights to them, which would diminish their effect on the joint likelihood, with a weight of zero, totally removing them. However, identifying such bad experts is not trivial without an error metric. It is also possible to mitigate their effects if most of the experts in the system are reliable. In this case, and if the number of experts is high, by combining all these experts, the impact of the bad ones would be reduced. In order to maximize robustness assigning an equal value to all models, as PoE does, may seem ideal as it minimizes the effect each individual likelihood has over the joint likelihood. However, it is important to understand




(a) PoE posterior




(b) Minimum entropy posterior

Fig. 5. While minimizing entropy yields better distributions in general, for certain cases, as the one illustrated (right), it fails. Failure occurs when a bad expert dominates the optimization, and generates a very confident, but wrong distribution. In such cases posteriors generated by the standard PoE approach (left) are considerably better.

that only informative likelihoods can mitigate the effects of bad experts. Consider a gPoE where experts generate the following likelihoods $l$, with Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$

$$l = \{\mathcal{N}(0,100), \mathcal{N}(0,100), \mathcal{N}(0,1), \mathcal{N}(0,1), \mathcal{N}(1,1)\}.$$

With $l_1$ and $l_2$ having much higher uncertainty than the remaining 3, and $l_5$ being a bad expert.

Now considering the following gPoE models, the first, $gPoE_1$, equally weights all experts, i.e., $\lambda_{1:5} = 1$; and the second, $gPoE_2$, disregards the highly uncertain models $l_1$ and $l_2$ and equally weights the remaining 3, i.e., $\lambda_{1,2} = 0$ and $\lambda_{3:5} = 1$. We obtain that the joint distributions are,

$$gPoE_1 = \mathcal{N}(0.331, 1.656)$$
$$gPoE_2 = \mathcal{N}(0.333, 1.0)$$

For this example it can be easily argued that $gPoE_2$ is a better posterior, as the effect of the bad expert is equally compensated as in $gPoE_1$, while maintaining an adequate variance. If maximum likelihood is used to evaluate the generated distributions, $gPoE_2$ has a higher probability with $gPoE_1(x=0) = 0.236$ and $gPoE_2(x=0) = 0.377$. In fact using maximum likelihood estimation, if it is not possible to discern among good and bad expert, $gPoE_2$ is ideal under the gPoE framework.

Taking inspiration from these ideas, we propose computing $\lambda$ using the gain in information from a uninformative prior to the posterior of locations conditioned to signal measurements. We compute this information gain as,

$$\lambda_j = H(\mathbf{x}) - H(\mathbf{x}|z_j), \tag{14}$$

with $H(\mathbf{x})$ being the entropy of the robot's location $\mathbf{x}$ and $H(\mathbf{x}|z_j)$ the conditional entropy of the location given the signal strength measurement $z_j$ from the access point $j$.

This measure completely removes models that generate uninformative posteriors, as they would be assigned a weight of zero. The weights for the example previously used to illustrate the effect of bad experts become $\lambda = \{0.0009, 0.0009, 0.9347, 0.9347, 0.9347\}$ (for $k = 20$ and $x \in [-5, 5]$), and the joint distribution $\mathcal{N}(0.333, 1.002)$, which is almost equal to $gPoE_2$ - considered ideal in our previous example.

We compute this measure by discretizing the location space into a grid $\{\mathbf{x}_i\}^{i=1:k}$ and assuming its mass point probabilities to be proportional to their probability density functions, so Eq. (14) becomes,

$$\lambda_j = \log(k) + \frac{1}{d}\sum_{i=1}^{k} p(z_j|\mathbf{x}_i)\log(p(z_j|\mathbf{x}_i)) - \log(d), \quad (15)$$

with $d = \sum_{i=1}^{k} p(z_j|\mathbf{x}_i)$.

As the weight $\lambda_j$ for the access point $j$ only depends on its measured signal strength $z_j$, it is tractable (memory wise) to compute Eq. (15) for all access points $j$ at several signal measurement values beforehand and cache them in memory for faster online evaluation of $\lambda$. Then, any observed signal strength $z$, can be interpolated from stored values in memory.

It is also possible to generalize our measure for its use with continuous variables. This avoids the discretization of the location space and, more interestingly, provides an analytical solution for weight computation. Although continuous implementations are computationally faster and can provide truly interesting and valuable insights, their derivation is often non-trivial.

### C. Gaussian Processes (gPoE GP)

As previously mentioned, it is also possible to learn the mapping $\mathbf{z} \rightarrow \lambda$, directly from training data. The main advantage of training models with recorded data is the possibility of employing error metrics, instead of metrics that solely rely on entropy. Error metrics refer to those that are derived from the comparison of the proposed solution to the ground truth solution for the particular example. The main advantage of using an error metric is that bad-experts can be easily identified and eliminated.

More specifically, the error metric chosen for this section is the cross entropy of the generated posteriors conditioned on measurements with respect to a prototype distribution, in this work, a bi-variate normal distribution centered on the ground truth location $\mathbf{x}_{gt}$ with a standard deviation $\sigma_{ce}$. That is:

$$CE(\mathbf{x}|\mathbf{z}) = \log(c) - \sum_{\mathbf{x}_i \in \mathcal{X}} \mathcal{N}(\mathbf{x}_i; \mathbf{x}_{gt}, \sigma_{ce}^2)\log(p(\mathbf{z}|\mathbf{x}_i)), \quad (16)$$

with $c = \sum_{\mathbf{x}_i \in \mathcal{X}} p(\mathbf{z}|\mathbf{x}_i)$. Lower cross entropy values are desired, as cross entropy can be understood as the weighted average of the model's negative log likelihood. By not only computing the likelihood of the ground truth, but also the likelihood of neighboring locations, a more robust metric is computed. For our case, the selected standard deviation for the prototype distribution, $\sigma_{ce}$, is set to 1.5 m, as posteriors are not expected to have lower standard deviations.

The main drawback of this approach is that error metrics can only be applied to labeled training data, which is sparse due to the high dimensionality of the mapping of interest. As data-driven approaches lack generalization outside their training inputs, the lack of large and representative enough training datasets is a big issue. To overcome this fragility against small training datasets we use the previously defined information gain as a prior over the data-driven model. By relying on this prior when no data is available, the model's outputs can be generalized outside its confined training input space. It can be considered that this proposed method enhances the previously proposed deltaH metric by using training data.

Specifically, we use Gaussian Processes (GPs), which can be used to learn complex mappings by defining a mean and a kernel function. Readers interested in GPs are referred to [19] for an in-depth description of this approach, and to [20] for an example of how priors over training data can enhance GP-based models.

To achieve this, we set the information gain function Eq. (15) as the GP's mean function. The mapping is then learned from training data using the same training data employed to learn the individual WiFi mappings, $(\mathbf{X}, \mathbf{Z})$ (with $\mathbf{X}$ being the matrix of the $n$ locations where the signals $\mathbf{Z}$ from the $m$ access points were acquired). The optimal weights for these data points are computed by minimizing Eq. (16), as

$$\Lambda = \underset{\lambda_i}{\operatorname{argmin}} \log(c) - \sum_{\mathbf{x}_k \in \mathcal{X}} \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{gt}, \sigma_{ce}^2) \sum_{j=1}^{m} \lambda_{i,j} \log(p(z_{i,j}|\mathbf{x}_k)), \quad (17)$$

for all $i \in 1:n$, with,

$$c = \sum_{\mathbf{x}_k \in \mathcal{X}} \exp\left(\sum_{j=1}^{m} \lambda_{i,j}\log(p(z_{i,j}|\mathbf{x}_k))\right). \quad (18)$$

Solving Eq. (17) using any convex optimization method (conjugate gradient descend in this work), yields the optimal weights matrix $\Lambda \in \mathbb{R}^{n\times m}$, with $n$ being the number of training points and $m$ the number of access points.

A GP is then learned using training data pairs $\{\mathbf{Z}, \Lambda\}$ with the information gain equation as its mean function ($m(\mathbf{z}) = H(\mathbf{x}) + H(\mathbf{x}|\mathbf{z})$). Once kernel parameters have been learned following the procedures described in [19], weights are computed given an arbitrary new measured signal strength vector $\mathbf{z}_{new}$, as the mean predicted value of the learned GP and considering the new input-outputs learned as well as the mean function, as,

$$\lambda = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}_{new}) + k(\mathbf{Z}, \mathbf{z}_{new})^T (k(\mathbf{Z}, \mathbf{Z}) + \sigma_m^2 \mathbf{I}_m)^{-1} (\Lambda - H(\mathbf{x}) + H(\mathbf{x}|\mathbf{Z})), \quad (19)$$

with $k(\cdot, \cdot)$ being the learned kernel function.

## IV. Experimental evaluations

### A. Testing environments and datasets

For the evaluation of our proposed approach, we surveyed three different buildings at the University of Tokyo, shown in Fig. 6. For all tests, a notebook was placed on top of a Pioneer 3 DX mobile robot with the notebook acquiring all signal strength measurements at each building, and the Pioneer robot providing mobility. Recorded data were: signal strengths, odometry, and laser rangefinder measurements; all recorded with timestamps in a rosbag and available online[1].

[1] http://www.robot.t.u-tokyo.ac.jp/~miyagusuku/software

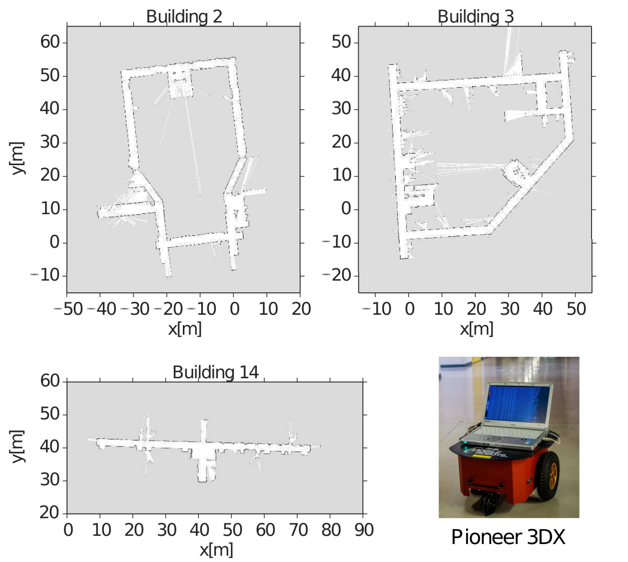
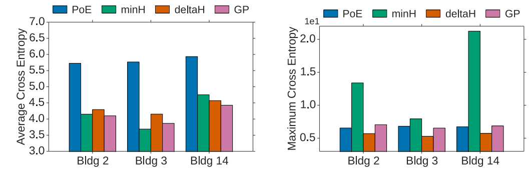



Fig. 6. Occupancy grid maps of the environments used for testing and the robotic platform employed.




(a) Average cross entropy (b) Maximum cross entropy

Fig. 7. Average and maximum cross entropy scores for all testing datasets using PoE, gPoE minH, gPoE deltaH and gPoE GP models.

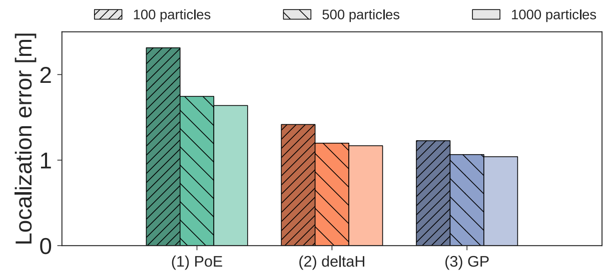



Fig. 8. Localization errors using different data information fusion approaches.

For all tested environments, an initial run with the robot was performed in order to collect the training dataset; and two additional testing datasets were obtained at a later date. The robot was operated continuously, without stopping, at speeds between 0.18 and 0.35 m/s.

Signal strength information was recorded only from beacon frames to guarantee that signals come from access points and not mobile users. Both signal strengths and odometry data are utilized in our localization algorithm, while range data is only used for building occupancy maps and obtaining datasets' ground truth locations (for both training and evaluations).

For all approaches, WiFi models were learned using the approach described in [6] and tested on two different datasets. Reported values are the average performance of these two testing datasets.

### B. Cross Entropy

To quantify the quality of the generated joint likelihood functions, we compute their cross entropy scores as in Eq. (16). Figure 7 shows the average and maximum cross entropy scores for the discussed models in this work: the standard PoE approach as well as all proposed gPoE models: minH, deltaH, and GP. Lower cross entropy scores indicate better joint posteriors as they would be closer to the ideal joint distribution. Models should not only have low average cross entropy scores but also low maximum cross entropy scores. Large maximum cross entropy values indicate that for some test inputs, generated likelihoods are not adequate. This can considerably decrease the reliability of the localization system using these distributions, hence it is important to avoid models that generate them.

As it can be observed from the figure, all gPoE models have lower average cross entropy scores, which means that on average, they generate better distributions. They also possess either similar or lower maximum cross entropies, which means they are as reliable as the classic PoE approach, with the exception of gPoE minH. As previously discussed, minimizing entropy is not a reliable way to compute joint posteriors due to bad experts. This experimental validation corroborates that such bad experts do occur (and frequently) in WiFi models. Hence, the use of gPoE minH is discouraged.

### C. Localization accuracy

To test the localization accuracy improvements by using the proposed models, we use them as the perceptual models of a standard Monte Carlo Localization (MCL) algorithm. MCL algorithms are a family of algorithms widely used for localization in robotics, that implements a Bayes filter, and are often the default choice given their ease of implementation and good performance across a broad range of localization problems - readers are referred to [6] for discussions on how to implement MCL using WiFi perceptual models.

Using the time-stamped data logs recorded, all evaluations are performed in a real-time manner using the open source framework ROS (Robot Operating System [21]). Localization errors are defined as the x-y Cartesian distance between the ground truths (as computed by laser rangefinders) and those estimated by WiFi. Figure 8 shows the average localization errors obtained using these different MCLs with 100, 500 and 1000 particles. Average localization errors were computed by averaging 50 runs of each testing configuration (testing dataset, perceptual model and number of particles).

As it can be observed there is a notable improvement in localization accuracy when the proposed gPoE models are used (gPoE minH was not tested, as its inadequacy was already concluded). It is particularly notable that gPoE deltaH is able to obtain localization accuracies comparable with gPoE GP although no training data is used for its computation.

## V. Conclusions and Future works

In this work, we have explored a vital, yet commonly overlooked step in WiFi-based localization: the generation of joint likelihood distributions based on the aggregation of individual likelihoods. To accomplish this, we have proposed the use of the general product of experts framework. This framework computes joint distributions as the weighted geometric average of individual likelihoods. To compute these weights several methods have been presented, discussed and compared. While this work has focused on the particular application example of WiFi-based self-localization, data information from multiple sources is an important subject for many applications, and it is our strong belief that the developed models can be applied to most of them.

Two methods based on information theoretic measures and that do not require training data have been introduced: gPoE minH and gPoE deltaH. gPoE minH casts weights computation as a minimization problem, while gPoE deltaH computes them based on the information gain from priors to posteriors when conditioning locations to sensor data. From the two, gPoE deltaH is preferred as experimental evaluations have shown that although gPoE minH had better average performance, it had very poor worst-case performance.

To take advantage of available training data, even if sparse, a third method based on a data-driven approach, named gPoE GP, has also been proposed. gPoE GP first computes ideal (in terms of minimal cross entropy) weights for each training data point available; then uses training data with their corresponding ideal weights to learn a signal strength to weights mapping using Gaussian Processes. Gaussian Processes effectively learn functions to compute weights and impose smoothness constraints (which improved model robustness to noise and worst-case performance). To handle the lack of dense training data, the previously proposed gPoE deltaH weights were used as priors over the GP. Hence, when data is available, the system tends to the computed ideal weights, while when no data is available, it tends to gPoE deltaH weights. This allows gPoE GP to have better average performance than gPoE deltaH while maintaining an acceptable worst case performance.

Experimental validations demonstrated the advantages of gPoE deltaH and GP over the classic PoE model, as well as the relation between better joint distributions, in terms of lower cross entropy scores, with better localization accuracies. Therefore, both gPoE deltaH and GP are strongly recommended over the classic PoE model. Among gPoE GP and gPoE deltaH, the latter is recommended when training or evaluation times are restricted/critical or when too few training data is available; if there are no such restrictions, gPoE GP is recommended as it consistently obtained better localization performance.

Future work will explore other data-driven methods which could take advantage of the underlying WiFi models, which are generative probabilistic models, hence could be used to generate as much synthetic data as necessary to train more complex models, as well as other applications.

## References


[1] F. Lemic, A. Behboodi, V. Handziski, and A. Wolisz, "Experimental decomposition of the performance of fingerprinting-based localization algorithms," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2014, pp. 355–364.

[2] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2010, pp. 4379–4384.

[3] J.-S. Gutmann, P. Fong, and M. E. Munich, "Localization in a vector field map," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 3144–3151.

[4] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian Process Positioning System for Cellular Networks." in *Advances in Neural Information Processing Systems*, 2003, pp. 579–586.

[5] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceedings of Robotics Science and Systems*, 2006, pp. 1–8.

[6] R. Miyagusuku, A. Yamashita, and H. Asama, "Precise and accurate wireless signal strength mappings using Gaussian processes and path loss models," *Robotics and Autonomous Systems*, vol. 103, pp. 134–150, 2018.

[7] ——, "Gaussian processes with input-dependent noise variance for wireless signal strength-based localization," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, Oct 2015, pp. 1–6.

[8] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models." in *IJCAI*, vol. 7, 2007, pp. 2480–2485.

[9] J.-S. Gutmann, E. Eade, P. Fong, and M. E. Munich, "Vector field slam - localization by learning the spatial variation of continuous signals," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 650–667, June 2012.

[10] S. Eisa, J. Peixoto, F. Meneses, and A. Moreira, "Removing useless aps and fingerprints from wifi indoor positioning radio maps," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–7.

[11] J. Y. Zhu, J. Xu, A. X. Zheng, J. He, C. Wu, and V. O. Li, "Wifi fingerprinting indoor localization system based on spatio-temporal (st) metrics," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2014, pp. 611–614.

[12] C. Beder, A. McGibney, and M. Klepal, "Predicting the expected accuracy for fingerprinting based wifi localisation systems," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2011, pp. 1–6.

[13] G. E. Hinton, "Products of experts," in *Proceedings of the Ninth International Conference on Artificial Neural Networks*, vol. 1, 1999, pp. 1–6 vol.1.

[14] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of gaussian process predictions," in *Proc. Mod. Nonparametrics 3: Autom. Learn. Pipeline workshop at NIPS*, 2014.

[15] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the American Control Conference*, vol. 4, 1997, pp. 2369–2373.

[16] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-gaussian dependence," in *Proceedings of the 15th International Conference on Information Fusion*, 2012, pp. 1876–1883.

[17] M. Uney, D. E. Clark, and S. J. Julier, "Information measures in distributed multitarget tracking," in *Proceedings of the 14th International Conference on Information Fusion*, 2011, pp. 1–8.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, MA: MIT press, 2005.

[19] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. Cambridge, MA: MIT press, 2006.

[20] R. Miyagusuku, A. Yamashita, and H. Asama, "Improving gaussian processes based mapping of wireless signals using path loss models," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 4610–4615.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *Proceedings of the ICRA Workshop on Open Source Software*, 2009, pp. 1–6.