Report 1

Position Estimating in Peer-to-Peer Networks

Authors:[1] Rogier Brussee[2], Mirela Dărău[3], Marta Dworczynska[4], Yabin Fan[3], Paulien Koeleman[5], Piotr Kowalczyk[6], Nico Schlömer[7], Tomasz Swist[8], Sandra van Wijk[3]

Abstract

We present two algorithms for indoor positioning estimation in peer-to-peer networks. The setup is a network of two types of devices: reference devices with a known location and blindfolded devices that can determine distances to reference devices and each other. From this information the blindfolded devices try to estimate their positions. A typical scenario is navigation inside a shopping mall where devices in the parking lot can make contact with GPS satellites, whereas devices inside the building make contact with each other, devices on the parking lot, and devices fixed to the building. The devices can measure their in-between distances, with some measurement error, and exchange positioning information. However, other devices might only know their position with some error.

We present two algorithms for positioning estimation in such a peer-to-peer network. The first one is purely geometric and is based on Euclidean geometry and intersecting spheres. We rewrite the information to a linear system, which is typically overdetermined. We use least squares to find the best estimate for a device its position. The second approach can be considered as a probabilistic version of the geometric approach. We estimate the probability density function that a device is located at a position given a probability density function for the positions of the other devices in the network, and a probability density function of the measured distances. First we study the case with a distance measurement to a single other user, then we focus on multiple other users. We give an approximation algorithm that is the probabilistic analogue of the intersecting spheres method. We show some simulated results where ambiguous data lead to well defined probability distributions for the position of a device. We conclude with some open questions.

1.1 Introduction

There are many wireless network applications for which knowing the location of the devices within the network is necessary. Think, for example, of a military or police operation using a radio network. Other examples are locating a specific car in a parking garage, or finding your seat in a large stadium.

[1] We would like to thank Jaron Samson from European Space Agency, The Netherlands, for useful discussions during the week. Thanks furthermore to the other participants who helped during the week: Rob De Staelen (University of Gent, Belgium), Rashid Mirzavand Boroujeni (TU Eindhoven, The Netherlands), and Valentina Masarotto (TU Delft, The Netherlands).

[2] Novay Institute, The Netherlands

[3] Technische Universiteit Eindhoven, The Netherlands

[4] Wroclaw University of Technology, Poland

[5] Vrije Universiteit Amsterdam, The Netherlands

[6] University of the West of Scotland, United Kingdom

[7] Univeristy of Antwerpen, Belgium

[8] National Bank of Poland, Poland

For outdoor situations GPS is widely used, and location estimates using GPS are in general very accurate [1]. However, for indoor use it is not possible to use GPS as the signal is quickly absorbed or blocked by walls. A number of indoor positioning technologies exist, see Zeimpekis et al. [2] for an overview.

In this paper we consider a network situation in which a small number of devices have knowledge of their locations, for example from using GPS or because their location is fixed. We call these devices *reference devices*. Most of the devices however, the so-called *blindfolded devices*, do not have this knowledge, but they wish to estimate their own location on the basis of location and distance information they exchange with other uses in the network. Such a network is known as a *peer-to-peer network*. We develop two algorithms for position estimation in these kind of networks. One is based on least squares methods and triangulation, and the other involves convolutions of probability density functions of the device locations.

The rest of the paper focuses on the two algorithms. Firstly, we explain the problem setting in more detail, and discuss the assumptions made in Section 2. In Section 1.3 we present the algorithm using the least squares methods. In Section 1.4 we give the algorithm based on probability density functions. Finally, in Section 1.5 we give our conclusions and suggestions for future research.

1.2 Problem description

We consider the following indoor peer-to-peer positioning problem. A number of customers inside a shopping mall want to know their positions inside the building. In order to estimate its position, each customer has a GPS device which measures the distances to and can exchange positioning information with other customers' devices. These customers can either be inside the building as well, or outside, e.g. on the parking lot. Customers inside the building might have connection to one or a few satellites, and hence might have some partial information about their exact positions. The customers outside have full satellite connection, and hence know their positions. We refer to these devices with known position as reference devices. This also includes fixed devices, such as routers, of which the position is known. The so-called blindfolded devices do not have exact position knowledge. They can try to estimate their own location based on location and distance information they exchange with other devices in the network. However, the distance measurements contain some error, and other (blindfolded) devices might be insecure about their positions as well.

We assume that the reference devices have exact or close to exact position estimates. This is because satellite connections give a location estimate with an error that is negligible compared to the error in location estimates made by blindfolded devices. The customers inside the mall are blindfolded devices, and we study the problem of how they can compute their locations, given that some of them have a connection with reference devices. We want to find an estimate for the location of each customer inside the mall.

We make the following simplifying assumptions. Firstly, we assume that there is no reflection of signal from the walls, or obstruction of the distance measurement is some way. Also, we assume the errors in the distance measurements between two devices, to be Gaussian distributed with zero mean. The variance in the error is the same for all devices. However, we do include the fact that signals die out over a distance. For this, we assume that two devices can exchange information only if they are closer to each other than some maximum distance, which is assumed equal for all device pairs. All devices within reach are referred to as the neighbours of a given device.

We focus on a two dimensional, static problem. So, we do not allow devices to change position. In consecutive time steps, each of the devices performs distance measurements to other devices within its range, and exchanges position information. From this information, the devices estimate their positions. The question is, how a device should best process the information. We want to find an estimate for the position, as well as the accuracy (error) of this estimate. For this, two algorithms are given in the next two sections. Firstly, we consider the problem after a single time step and give an algorithm using least squares estimates. Secondly, we give an algorithm based on probability density functions, focusing also on multiple time steps.

1.3 Least squares algorithm

In this section, we give a method for estimating a given device its position after a single time step. Let X, with coordinates (x, y), be the unknown position of the blindfolded device under consideration. Suppose it is within range of n neighbours, say A_i for $i = 1, \ldots, n$, with (estimated) coordinates (x_i, y_i). This estimate is the actual location plus some error δ_i. Also, the outcome of a single distance measurement between X and A_i is known, for all i. These measurements consists of the actual distance plus some error ε_i, which we assume to have a Gaussian distribution with zero mean and standard deviation σ. From this information, we want to estimate the position X of the device.

We start by abstracting the problem in geometrical terms in the following way. From the coordinates (x_i, y_i) and the error δ_i, it follows that the exact position of A_i should be contained in a disc with radius δ_i around (x_i, y_i). That is, the exact position of A_i is contained in the following domain:

$$\{(a_{i\,x}, a_{i\,y}) | (a_{i\,x} - x_i)^2 + (a_{i\,y} - y_i)^2 \leq \delta_i^2\}, \quad \forall i = 1, \ldots, n. \tag{1.1}$$

The exact position of X is contained in the domain:

$$\{(x, y) | (d_i - |\varepsilon_i|)^2 \leq (x - a_{i\,x})^2 + (y - a_{i\,y})^2 \leq (d_i + |\varepsilon_i|)^2\}, \tag{1.2}$$

which is a ring centered at $(a_{i\,x}, a_{i\,y})$, with inner radius $d_i - |\varepsilon_i|$ and outer radius $d_i + |\varepsilon_i|$. From (1.1) and (1.2), we have that X is in:

$$\{(x, y) | (d_i - \delta_i - |\varepsilon_i|)^2 \leq (x - x_i)^2 + (y - y_i)^2 \leq (d_i + \delta_i + |\varepsilon_i|)^2\}, \quad \forall i = 1, \ldots, n. \tag{1.3}$$

This is again a ring, now centered at (x_i, y_i) with radii $d_i \pm (\delta_i + |\varepsilon_i|)$. The probabilities of $\varepsilon_i \in (-\sigma, \sigma)$, $(-2\sigma, 2\sigma)$, respectively $(-3\sigma, 3\sigma)$ are about 68.2%, 95.4%, respectively

99.6%. Now based on geometrical considerations, X should be inside the intersection of the n domains given by (1.3):

$$\bigcap_{i=1}^{n} \{(x,y)|(d_i-\delta_i-|\varepsilon_i|)^2 \leq (x-x_i)^2+(y-y_i)^2 \leq (d_i+\delta_i+|\varepsilon_i|)^2\}. \tag{1.4}$$

Example 1.3.1. Consider the following example for $n=4$, where

$$\begin{aligned}\{(x_i,y_i,\delta_i,d_i) \mid i=1,\dots,4\} = \{&(1,1,0.2,\sqrt{2}),(3,1,0.01,\sqrt{10}),\\ &(-2,0,0.08,2),(3,-3,0.1,3\sqrt{2})\},\end{aligned}$$

$\sigma = 10^{-2}$ and $\varepsilon_i \in (-3\sigma, 3\sigma)$ randomly chosen. Figure 1.1 shows the four rings specified by (1.3). The position of X should be inside the intersection of these four rings. When zooming in, Figure 1.2 gives a closer look of the intersection (the shadowed part), i.e. the domain that X belongs to, which is (1.4).

Remark 1.3.2. When there are no errors in the distance measurements and the positions (x_i,y_i), i.e. when all $\delta_i, \varepsilon_i = 0$, then domain (1.4) reduces to a single point.

Remark 1.3.3. The more information available (i.e. the larger n), the smaller the domain (1.4) is, i.e. the more accurately X's position is estimated.

Now we know that X should be inside a given domain, the question is which point inside this domain should be chosen as a best estimate for X. In the following, let us consider the problem in a different way. Knowing the outcomes of the n distance measurements to its neighbours, we have the following system of equations for (x,y):

$$\begin{aligned}(x-x_1)^2+(y-y_1)^2 &= d_1^2\\ (x-x_2)^2+(y-y_2)^2 &= d_2^2\\ &\vdots\\ (x-x_n)^2+(y-y_n)^2 &= d_n^2\end{aligned} \tag{1.5}$$

Since the system is quadratic, and therefore hard to solve, we use the following procedure to reduce it to a linear system. Let

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_i, \quad \overline{y} = \frac{1}{n}\sum_{k=1}^{n} y_i, \quad \overline{d}^2 = \frac{1}{n}\sum_{k=1}^{n} d_i^2. \tag{1.6}$$

Then, for $i=1,\dots,n$:

$$(x-x_i)^2 = (x-\overline{x})^2+(\overline{x}-x_i)^2+2(x-\overline{x})(x-x_i), \tag{1.7}$$

$$(y-y_i)^2 = (y-\overline{y})^2+(\overline{y}-y_i)^2+2(y-\overline{y})(y-y_i). \tag{1.8}$$

Summing (1.7) and (1.8) over all $i=1,\dots,n$, and using the notation of (1.6), we have

$$(x-\overline{x})^2+(y-\overline{y})^2 = \frac{1}{n}\sum_{k=1}^{n} d_i^2 = \overline{d}^2. \tag{1.9}$$

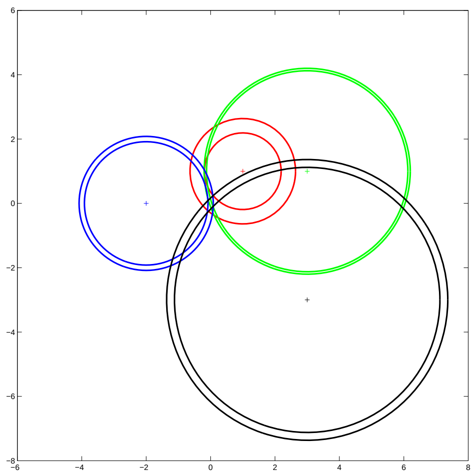


Figure 1.1: Figure of Example 1, for four neighbours.

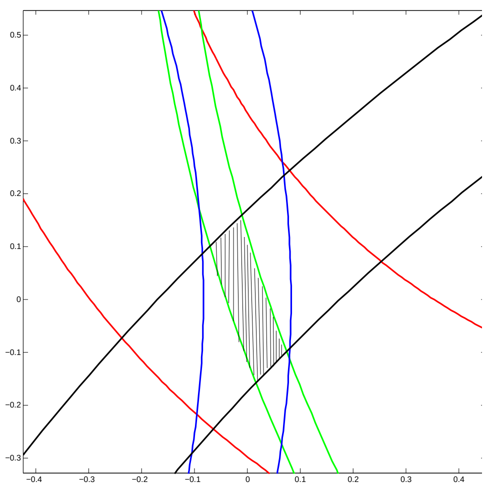


Figure 1.2: Again Figure 1.1, now zoomed in at the intersection.

Substituting (1.7) and (1.8) in (1.5), and using (1.9), we obtain the linear system

$$
\begin{aligned}
(x_1-\overline{x})x+(y_1-\overline{y})y &= \frac{1}{2}\left(d_1^2-d^2+(\overline{x}^2+\overline{y}^2)-(x_1^2+y_1^2)\right)\\
(x_2-\overline{x})x+(y_1-\overline{y})y &= \frac{1}{2}\left(d_2^2-d^2+(\overline{x}^2+\overline{y}^2)-(x_2^2+y_2^2)\right)\\
&\vdots\\
(x_n-\overline{x})x+(y_1-\overline{y})y &= \frac{1}{2}\left(d_n^2-d^2+(\overline{x}^2+\overline{y}^2)-(x_n^2+y_n^2)\right)
\end{aligned}
\tag{1.10}
$$

This system is generally over-determined, therefore we use the least squares method to determine the best estimate for (x, y). That is, when writing (1.10) as $AX = b$, we find $X = (x, y)$ such that

$$||AX - b||_2$$

is minimized. Here $||\cdot||_2$ denotes the Euclidean norm, defined by $||z||_2 = \left(\sum_{i=1}^n z_i^2\right)^{1/2}$, for some real-valued vector $z = (z_1, \ldots, z_n)$. Hence, the best position estimate for X is the least square solution of the system (1.10).

1.4 Probability density functions algorithm

In this section we present a holistic approach to the uncertainty of localization of clients. The idea is that at each moment, client k's notion of its absolute position is described by a random variable X_k and an associated probability density function (pdf) ρ_{X_k}. This concept captures other more specific approaches, such as the one described in Section 1.3, which can be embedded into the present approach by associating each client k with pdf

$$
\rho_k(\boldsymbol{x}) = \begin{cases} (\pi r^2)^{-1} & \text{if } \|\boldsymbol{x}-\boldsymbol{x}_k\|_2 \le r, \\ 0 & \text{otherwise,} \end{cases}
$$

where $\boldsymbol{x}_k$ is the true position of client k, and r is the range of the distance measurement. In this sense, the present methodology aims to provide a general framework with the help of which it can be inferred what localization information generally can and cannot be deducted from a given client configuration. In this sense, all more simplistic approaches could use this approach as a reference framework.

The general idea is that in each (time) step, a client's pdf is updated based upon all available information, that is, measurements of its distance to neighbouring clients and their respective pdfs. In this way, within a group of clients, every individual client can alter and hopefully improve its localization pdf over time.

We try to outline in how far one can defer properties of the clients pdf by distance measurements to other clients and their pdfs. We focus on the two-dimensional case, although all presented ideas can be formulated effortlessly in any spatial dimension.

In Subsection 4.1 some required notation is introduced and the model is laid out in greater detail. Then, in Subsection 4.2, the focus is on the mathematical steps that a client has to perform to build its own pdf. This is done based upon measurements with respect to other clients. Subsection 4.3 contains numerical experiments with simple client setups from which suggestions are derived about the characteristics of possible scenarios. Finally, the results are discussed in Subsection 4.4.

Model and notation

When a client tries to determine its position based on measurements to other clients, there are two kinds of uncertainties involved: the distance measurement itself and other clients' pdfs. We capture these uncertainties in two random variables, which are both introduced below.

Position. Let the generally unknown true position of client k be denoted by $\boldsymbol{x}_k \in \mathbb{R}^2$, and let the random variable X_k describe the assumed position of client k. Let ρ_k denote the corresponding probability density function. Hence, the probability that client k is located in $\Omega \subseteq \mathbb{R}^2$ is given by $P(\boldsymbol{x}_k \in \Omega) = \int_\Omega \rho_k(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$. The random variable X_k may typically be normally distributed with mean $\boldsymbol{x}_k$, but in general, ρ_k can be every normalized integrable function.

Distance measurement. Let $D^{1\mathrm{D}}_{k_0,k_1}$ be the random variable giving the distance measurement of one client k_0 to another client $k_1 \neq k_0$. It is assumed that $D^{1\mathrm{D}}_{k_0,k_1}$ is normally (i.e. Gaussian) distributed with (unknown) mean $d_{k_0,k_1} := \|x_{k_0} - x_{k_1}\|_2$ and a given standard deviation $\sigma > 0$. This standard deviation could e.g. be a property of the devices in use. So,

$$\begin{aligned} D^{1\mathrm{D}}_{k_0,k_1} &\sim \mathcal{N}(d_{k_0,k_1}, \sigma^2), \\ \rho_{D^{1\mathrm{D}}_{k_0,k_1}}(x) = \rho_{\mathcal{N}}(x; d_{k_0,k_1}, \sigma) &:= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2}\left(\frac{x - d_{k_0,k_1}}{\sigma}\right)^2 \right\}, \end{aligned} \tag{1.11}$$

see Figure 1.3(a).

Based upon n statistically independent samples $\{D_i\}_{i=1}^n$ of $D^{1\mathrm{D}}_{k_0,k_1}$, it is the goal to estimate (1.11), that is, to estimate d_{k_0,k_1}. The most natural approach is here to use the maximum likelihood estimator

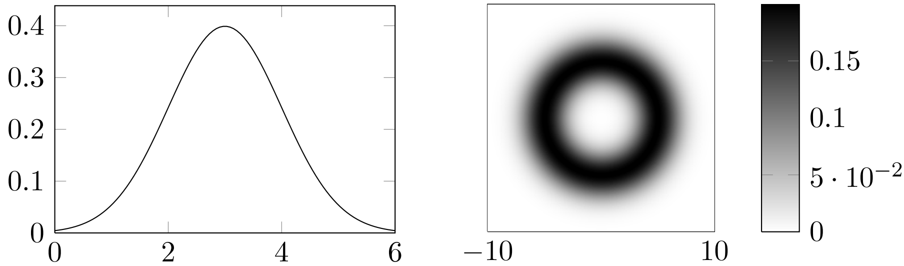
$$\widehat{D} := \frac{1}{n}\sum_{i=1}^{n} D_i$$

which is well known to be normally distributed as well, with standard deviation $\sigma/\sqrt{n}$:

$$\widehat{D} \sim \mathcal{N}(d_{k_0,k_1}, \sigma^2/n). \tag{1.12}$$

Based upon this, it is possible to determine the probability density function of the random variable $D^{2\mathrm{D}}_{k_0,k_1}$ which describes the location $\boldsymbol{d} := \boldsymbol{x} - \boldsymbol{x}_0$ relative to a fixed spot $\boldsymbol{x}_0$ to which a distance measurement according to (1.11) was done. As there is no preference in a particular spatial direction, the pdf associated with $D^{2\mathrm{D}}_{k_0,k_1}$ is given by

$$\rho_{D^{2\mathrm{D}}_{k_0,k_1}}(\boldsymbol{d}; \hat{d}, \sigma) = \frac{1}{2}\left(\frac{1}{2\pi}\rho_{D^{1\mathrm{D}}_{k_0,k_1}}\left(\|\boldsymbol{d}\|_2\,; \widehat{D}, \sigma\right) + \frac{1}{2\pi}\rho_{D^{1\mathrm{D}}_{k_0,k_1}}\left(\|\boldsymbol{d}\|_2\,; -\widehat{D}, \sigma\right)\right); \tag{1.13}$$



(a) Normal distribuion of $D^{1\mathrm{D}}_{k_0,k_1}$ with $d_{k_0,k_1} = 3$ and $\sigma = 1$.

(b) Ring distribution of $D^{2\mathrm{D}}_{k_0,k_1}$ with $d_{k_0,k_1} = 5$.

Figure 1.3: Assumed normal distribution of $D^{1\mathrm{D}}_{k_0,k_1}$ as defined in (1.11), along with the corresponding two-dimensional ring distribution of $D^{2\mathrm{D}}_{k_0,k_1}$ as defined in (1.13).

see Figure 1.3(b).

Note 1.4.1. It would also be possible to generically discard negative distance measurements, and instead of (1.11) assume a cut-off normal distribution

$$\tilde{\rho}_{D^{1\mathrm{D}}_{k_0,k_1}}(x) = \begin{cases} I^{-1}\rho_{\mathcal{N}}(x; d_{k_0,k_1}, \sigma) & \text{for } x > 0, \\ 0 & \text{otherwise} \end{cases}$$

for $D^{2\mathrm{D}}_{k_0,k_1}$. The normalization factor $I := \int_{x>0} \rho_{\mathcal{N}}(x; d_{k_0,k_1}, \sigma)$ is well-known not to have an analytic representation. However, this would lead to the simpler expression

$$\tilde{\rho}_{D^{2\mathrm{D}}_{k_0,k_1}}(\boldsymbol{d}; \hat{d}, \sigma) = \frac{1}{2\pi I}\rho_{D^{1\mathrm{D}}_{k_0,k_1}}\left(\|\boldsymbol{d}\|_2 ; \hat{d}, \sigma\right).$$

Note that $\tilde{\rho}$ is non-smooth at $\boldsymbol{x}_0$, which is in the context of pdfs not a restriction.

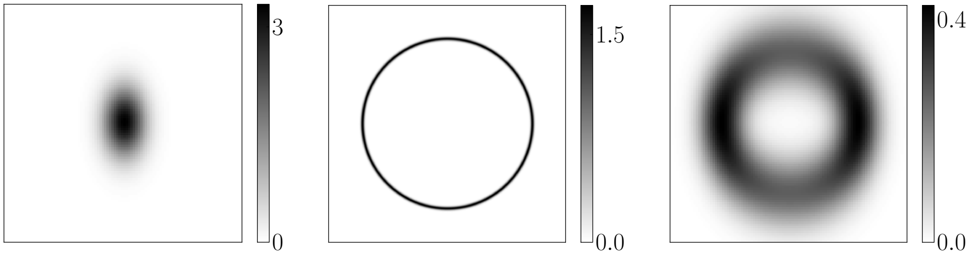
Deferring information on the location from other clients

We now make use of the knowledge gained by distance measurements to several neighbouring clients, and combine the uncertainties given by (1.13) with the pdfs of the respective neighbouring clients to get an estimate for the user's own pdf.

Distance measurements to one other client

The first question to be answered is how the estimated $\rho_{D^{2\mathrm{D}}_{k_0,k_1}}$ of (1.13) and $\rho_{X_{k_1}}$ can be cast into an approximation for $\rho_{X_{k_0,k_1}}$. As $\boldsymbol{x}_{k_0} = \boldsymbol{x}_{k_1} + \boldsymbol{d}$, we set $X_{k_0,k_1} := X_{k_1} + D^{2\mathrm{D}}_{k_0,k_1}$, where X_{k_0,k_1} describes the position of client k_0 with respect to communication with client k_1. Furthermore, the random variables X_{k_1} and $D^{2\mathrm{D}}_{k_0,k_1}$ are assumed to be independent, which means that the distance measurement (error) does not depend on the actual distribution of X_{k_1}. Hence,

$$\rho_{X_{k_0,k_1}} = \rho_{X_{k_1}+D^{2\mathrm{D}}_{k_0,k_1}} = \rho_{X_{k_1}} * \rho_{D^{2\mathrm{D}}_{k_0,k_1}},$$



(a) Normal distribution centered at the origin, with a slightly larger deviation in the y-direction than in the x-direction.

(b) Ring distribution as defined in (1.13) with deviation significantly smaller than the deviations of the normal distribution of Figure 1.4(a).

(c) Result of the convolution of the two distributions of Figures 1.4(a) and 1.4(b).

Figure 1.4: Illustration of the convolution process of (1.14). Figure 1.4(a) shows the convolution data, Figure 1.4(b) the convolution kernel, and Figure 1.4(c) the actual convolution of the two. Note how the smaller deviation in the x-direction of the bivariate Gaussian distribution increases the values of the convolution at the x-extreme ends of the localization pdf in (c).

where the asterisk denotes convolution, i.e.,

$$\rho_{X_{k_0,k_1}}(\boldsymbol{x}) = \iint_{\mathbb{R}^2} \rho_{X_{k_1}}(\boldsymbol{y}) \rho_{D^{2D}_{k_0,k_1}}(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \tag{1.14}$$

where $\rho_{X_{k_1}}$ is occasionally referred to as *convolution data* and $\rho_{D^{2D}_{k_0,k_1}}$ as *convolution kernel*. This expresses the notion that the 'data' distribution $\rho_{X_{k_1}}$ is acted upon and blurred by convolution with the 'kernel' $\rho_{D^{2D}_{k_0,k_1}}$. Note, though, that mathematically there is no distinction between data and kernel as the convolution is commutative.

The integral in (1.14) can be calculated numerically (e.g., using Fast Fourier Transformation). Figure 1.4 shows an example.

Example 1.4.2. Suppose that client k_1 knows its position exactly. So, its distribution is the Dirac-distribution centered at $\boldsymbol{x}_{k_1}$, that is $\rho_{X_{k_1}} = \delta_{\boldsymbol{x}_{k_1}}$. Then

$$\rho_{X_{k_0,k_1}}(\boldsymbol{x}) = \iint_{\mathbb{R}^2} \delta_{\boldsymbol{x}_{k_1}}(\boldsymbol{y}) \rho_{D^{2D}_{k_0,k_1}}(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \rho_{D^{2D}_{k_0,k_1}}(\boldsymbol{x}-\boldsymbol{x}_{k_1}),$$

i.e., the 'ring' distribution centered at $\boldsymbol{x}_{k_1}$. As expected, the only contribution to the uncertainty of X_{k_0,k_1} is the uncertainty rooted in the distance measurement itself.

Distance measurements to multiple other clients

Once all $X_{k_0,k}$ for all n neighbouring clients $k \in \{k_1, \dots, k_n\}$ have been calculated, the information contained in each of them is to be combined to a common pdf X_{k_0} that indicates the localization likelihood of client k_0 based upon measurements to all

reset own P to uniform distribution over Ω;
for *all clients* k_i *within reach* **do**
    measure the distance to k_i;
    create the corresponding ring distribution R_{k_i} using (1.13);
    request the pdf P_{k_i} of client k_i;
    $P \leftarrow P \cdot (P_{k_i} * R_{k_i})$;
end
normalize P;

Algorithm 1: One update step for the pdf of a client. In this algorithm information about previous updates is discarded. However, a client could very well store data of the previous measurements to decrease the uncertainty of its neighbours' positions. For the computations in the present paper, the mean of all previous computations is stored in each step and updated accordingly for the new measurement. This assumes that none of the clients changes its position throughout the process.

its neighbours. Quite naturally one could ask for the probability that $\boldsymbol{x}_0$ is contained in $\Omega \subset \mathbb{R}^2$ according to X_{k_0,k_1} *and* X_{k_0,k_2}. That is

$$P(X_{k_0,k_1} \in \Omega \wedge X_{k_0,k_2} \in \Omega) = P(X_{k_0,k_1} \in \Omega) \cdot P(X_{k_0,k_2} \in \Omega \mid X_{k_0,k_1} \in \Omega).$$

Assuming independence of X_{k_0,k_1} and X_{k_0,k_2}, this yields

$$P\left(X_{k_0,k_1} \in \Omega \wedge X_{k_0,k_2} \in \Omega\right) = P(X_{k_0,k_1} \in \Omega) \cdot P(X_{k_0,k_2} \in \Omega),$$

and subsequently for all $k \in \{k_1, \ldots, k_n\}$:

$$P\left(\bigwedge_{i=1}^{n} X_{k_0,k_i} \in \Omega\right) = \prod_{i=1}^{n} P(X_{k_0,k_i} \in \Omega).$$

This results in the pdf

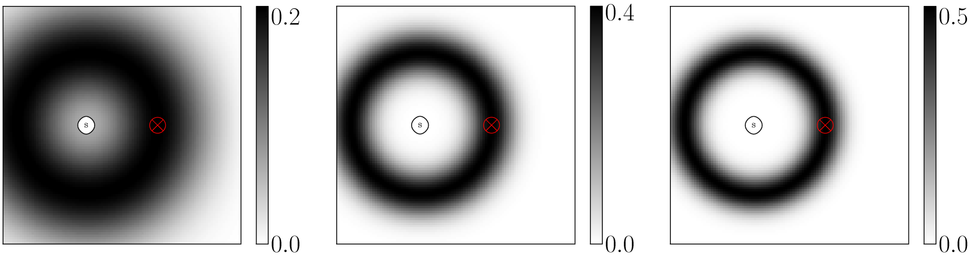
$$\rho_{X_{k_0}}(\boldsymbol{x}) = I^{-1} \prod_{i=1}^{n} \rho_{X_{k_0,k_i}}(\boldsymbol{x}),$$

with the normalization constant $I \in \mathbb{R}$.

Simulation

In this section, a few example constellations are set up and iterated over a number of time steps, to see how the pdfs of the individual users evolve when more and more accurate data of the other users becomes available. It is assumed that none of the clients moves during the process.

Each of the following example setups consists of a number of free clients ($\otimes$), having blindfolded devices. These are located in a square-shaped domain Ω with edge



(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.5: Experiment with one free client ($\otimes$) and one satellite client (Ⓢ) after multiple executions of Algorithm 1. The exact positions of the clients are depicted, as well as the pdf of the free client. After the first measurement, the free client can only determine its position up to rotation around the satellite client. Over time, the width of the rings becomes smaller as the deviation of the distance estimation diminished (see (1.12)), but with only one reference client the position cannot be further localized. The fuzziness of the ring distribution after 100 steps mainly depends on σ_{sat} rather than the uncertainty about the distance measurements.

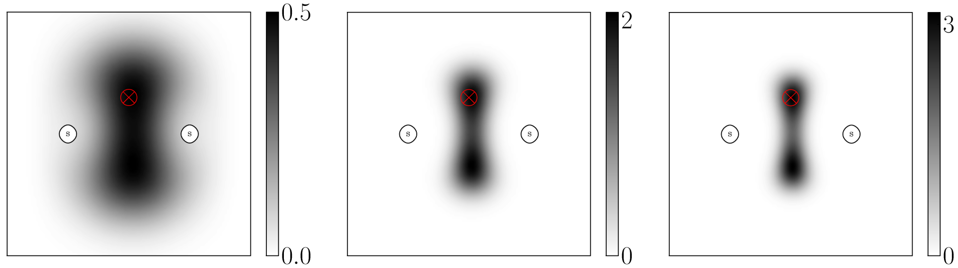
length normalized to 1.0. All values indicating lengths, including standard deviations of distance measurements, hence are in units of the domain edge length.

Every client k starts off with an insignificant pdf corresponding to the random variable X_k of its spatial position, i.e. the uniform distribution over the domain Ω. Furthermore, there is at least one client who already has a significant pdf from the start, which are modeled here by clients (Ⓢ) who have a connection to a set of GPS satellites and can thus determine their positions independently from other clients. Hence, these are the reference devices. It is assumed that the pdf of client k_{sat} is the normal distribution centered at its true position $\boldsymbol{x}_{k_{\mathrm{sat}}}$ with $\sigma_{\mathrm{sat}} = 0.05$ independently for all satellite clients.

With this given setup, each of the free clients now iteratively executes Algorithm 1 to update its pdf given pdfs of its direct neighbours. The simulation is run for six settings, see Figures 1.5 to 1.10, starting from a simple case with only one satellite client and one free client, ending up with a network of multiple clients of both types. We depict the pdfs of the indicated free client after 1, 10, and 100 steps. Typically, the probability density becomes more concentrated about one location or possibly multiple locations. In this way the most likely location(s) of the clients become clearer and clearer. For each setting we discuss the setup and the results.

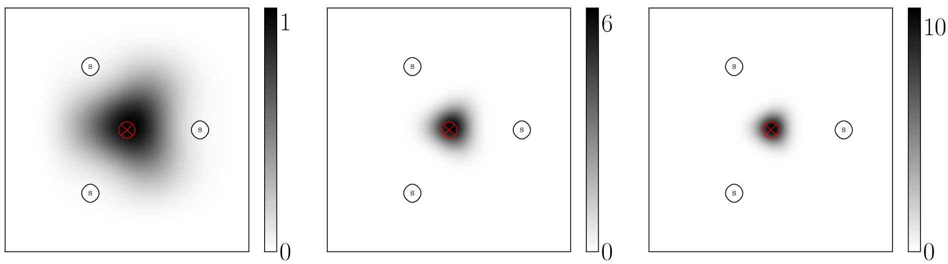
Discussion pdf algorithm

The presented approach gives in multiple settings reasonable outcomes, where the peak of the pdf coincides with the true position of the client. For some setups it was not possible to get any more specific information than a localization up to two significant spots (see for example Figure 1.9), although all available information was



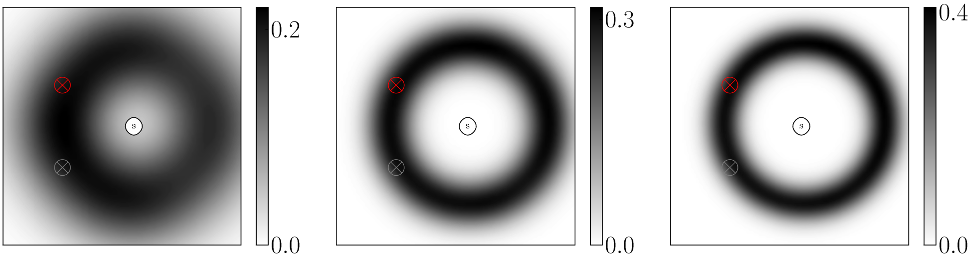
(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.6: Experiment with one free client (⊗) and *two* satellite clients (ⓢ) after multiple executions of Algorithm 1. After measurements to both of the satellite clients, the free client can be determined to be more likely in those regions where the two fuzzy rings overlap, i.e., where their product is locally maximal. The principal fact that two spots are preferred cannot be overcome, although again the uncertainty about the distance measurements is filtered out by sampling over 100 steps.



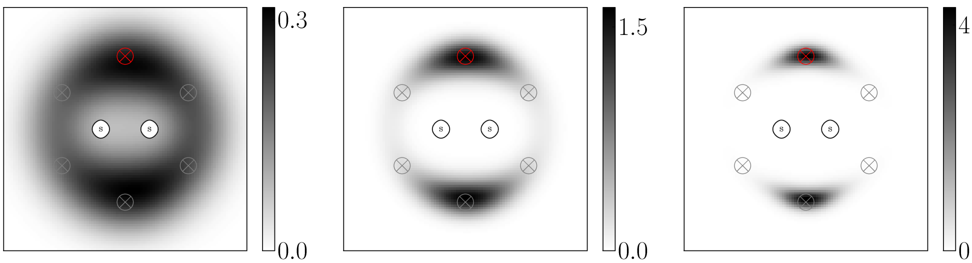
(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.7: Experiment with one free client (⊗) and *three* satellite clients (ⓢ) after multiple executions of Algorithm 1. As opposed to Figures 1.5 and 1.6, after one measurement already *one* fuzzy spot can be determined to likely contain the client's position. Again, certainty is increased as more samples in distance measurement are taken, such that the uncertainty σ_{sat} ultimately dominates for the free client as well.



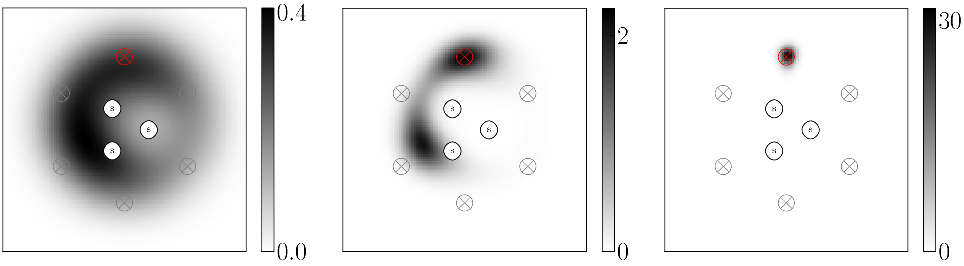
(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.8: Experiment with two free clients (⊗) and one satellite client (Ⓢ) after multiple executions of Algorithm 1. This situation is not much different as compared to the one in Figure 1.5, except for the fact that now two clients with (initially) rather uncertain information about their position can try to improve their localizations by exchanging information about their position relative to each other. It appears, though, that this extra information does not improve the position estimate of the individuals. The determining factor remains that there is only *one* client with absolute information about its position. This experiment suggests that adding clients without information on their absolute positions does not alter the uncertainty of localizations of present clients.



(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.9: Experiment with six free clients (⊗) and two satellite clients (Ⓢ) after multiple executions of Algorithm 1. No single free client has contact with both of the satellite clients, but still the localization information propagates through the network to each of the clients, such that they all have pdfs similar to the situation in Figure 1.6 rather than Figure 1.5.



(a) Step 1. (b) Step 10. (c) Step 100.

Figure 1.10: Experiment with six free clients (⊗) and three satellite clients (Ⓢ) after multiple executions of Algorithm 1. Quite curiously, the pdf of ⊗ has a curved shape with a preferred direction in space after one step. This is likely related to the particular (not fully rotational symmetric) arrangement of the satellite clients within the circle of the free clients. After a few steps, two spots of higher probability of localization are formed which is rather similar to the much simpler situation of Figure 1.6. One of the two preferred spots is indeed the true position of the client. After a few more steps, this location is actually preferred over the other local maximum, such that the client gets a good estimate about its position.

made use of. This means that, principally, no algorithm that uses less information can localize the client any more precisely.

An option for improvement still might be taking into account previous measurements by weighting them with the current one, and using this for the updating of the pdf. Currently, only the most recent measurement is taken into account, which can, by chance, have a large error. In an early stage of the updating process, this might cause the pdf to concentrate about an erroneous location, far off the true location, from which it needs many extra measurements to make it more accurate again. On the other hand, putting too much weights on previous measurements makes it last longer before the pdf concentrates around a location.

It must be noted that, for practical implementations, the given approach requires relatively heavy inter-client communication between the devices. Also, numerical calculation of the convolutions (1.14) is computationally expensive. This holds particularly for very fine-grained discretizations of the client's environment. An alternative for communicating an entire pdf is approximating it by, e.g., a bivariate Gaussian distribution. As only the few real-valued parameters have to be exchanged in this case, this would dramatically reduce the amount of data to be transferred. On the other hand, a bivariate Gaussian distribution might not always be suited for representing a client's position, see for example Figure 1.5. It is left for further research to investigate how much this decline in performance is, and what would be the best choice for approximation of a client's pdf.

1.5 Conclusion and discussion

In this paper we presented two promising methods for solving the problem of computing device locations from information about neighbour locations. The first method uses least square techniques, and is simple to implement and execute on a device. The second method uses complete information in the form of probability density fuctions and so gives more, and even more accurate, information than the first method. However, this comes at the cost of having to exchange more information between devices and using a more computation-intensive algorithm.

There are many posibilities for further research. A first important point for both methods is, if and how the algorithms converge. From our experiments we are confident that both methods converge to a location or probability distribution of a location for every device in the network, but that does not necessarily mean that they converge to the correct position. The conditions under which they actually do so, would be interesting, as would be the rate of convergence.

For the method using least squares, the question also remains how accurate the proposed method is, and how the error in the best estimated position can be determined. For the second method, an important issue still open, is the question if the amount of information transmitted and the computation to be done, can be reduced to make it more suitable for practical use. This reduction should be done with minimal loss in performance.

Finally, another important point for consideration is how these algorithms can be extended to include moving devices, that is, customers actually walking around in a mall. This makes the questions on speed of convergence and ease of computation even more relevant.

1.6 References

[1] M. Chivers. Differential GPS explained. *Environmental Systems Research Institute, Inc.*, 2003.

[2] V. Zeimpekis, G.M. Giaglis, and G. Lekakos. A taxonomy of indoor and outdoor positioning techniques for mobile location services. *ACM SIGecom Exchanges*, 3(4):19–27, 2002.