(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 September 2010 (23.09.2010)

PCT

(10) International Publication Number
WO 2010/106530 A2

(51) International Patent Classification:
G01S 5/02 (2010.01)

(21) International Application Number:
PCT/IE2010/000014

(22) International Filing Date:
18 March 2010 (18.03.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2009/0206 19 March 2009 (19.03.2009) IE

(71) Applicant *(for all designated States except US)*: **CORK INSTITUTE OF TECHNOLOGY** [IE/IE]; Bishopstown, County Cork (IE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KLEPAL, Martin** [CZ/IE]; 31A mount Eden, Model Farm Road, Bishopstown, County Cork (IE). **WIDYAWAN** [ID/IE]; c/o Cork Institute of Technology, Bishopstown, County Cork (IE). **PESCH, Dirk** [DE/IE]; Fairview, Fennell's Bay, Crosshaven, County Cork (IE).

(74) Agents: **O'BRIEN, John, A.** et al.; c/o John A. O'Brien & Associates, Third Floor, Duncairn House, 14 Carysfort Avenue, Blackrock, County Dublin (IE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: A LOCATION AND TRACKING SYSTEM

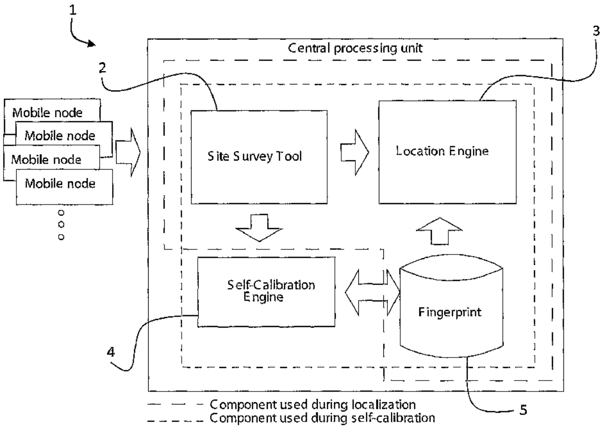


Fig. 1

(57) **Abstract**: A location and tracking system (1) comprises a site survey tool (2), a location engine (3), a self- calibration engine (4), and a fingerprint database (5). The system (1) operates for a wireless network of access points in a building and it generates from initialization data a model of the physical environment. The model may for example be in the form of data representing geometry of lines in plan and signal attenuation values for the lines. The system then uses this model to generate a fingerprint of projected signal strengths in the environment. The system may maintains a fingerprint for every environment model or there may be only one model at any one time, or there may be a number. Where the latter, the system may assign a weighting to each model to indicate its confidence value. Upon detection of a mobile node in the environment the system (l)estimates position of the node according to the current fingerprint, and in real-time performs self-calibration of the environment model, in turn giving rise to fresh updated fingerprints for future tracking. This continuous self-calibration ensures excellent tracking accuracy even with modification of the environment, such as changing of an access point configuration, moving a partition wall, or movement of a metal filing cabinet.

"A Location and Tracking System"

INTRODUCTION

Field of the Invention

The invention relates to location and tracking systems such as those within a building. An example is a system for navigation in large office buildings, for tracking location of security personnel in an airport.

In a typical location and tracking system the position of a mobile node is inferred from location-specific physical parameters of the wireless communication channel which exists between fixed nodes and the mobile node such as RSS (Radio Signal Strength), TDOA (Time Difference of Arrival), impulse response characteristics, angle of arrival, and multi-path propagation characteristics.

However, the wireless channel parameters in a multi-path environment (such as indoor environment) are subject to the electromagnetic properties of the environment such as signal transition and reflection losses, signal phase shift, and signal depolarisation when interacting with the environment.

The invention is directed towards providing a location and tracking system with improved ability to track locations of mobile nodes. Another objective is to avoid need for considerable additional equipment in the environment.

SUMMARY OF THE INVENTION

According to the invention, there is provided a location and tracking system comprising wireless base stations, an input interface, and a processor, wherein the processor is adapted to

(a) receive physical environment inputs and provide an initial physical environment model;

(b) generate an initial channel parameters fingerprint for the environment according to the initial environment model;

(c) monitor channel parameters of a mobile node in the environment and use said parameters and the fingerprint to locate the mobile node;

(d) modify the environment model according to said node locating step (c);

(e) update the fingerprint according to the modification of the environment model in step (d); and

(f) repeat steps (c) to (e) in subsequent iterations.

In another aspect, the invention provides a location and tracking method performed by a location and tracking system linked with wireless base stations, and having an input interface, and a processor, wherein the method comprises the steps of:

(a) receiving physical environment inputs and providing an initial physical environment model;

(b) generating an initial channel parameters fingerprint for the environment according to the initial environment model;

(c) monitoring channel parameters of a mobile node in the environment and using said parameters and the fingerprint to locate the mobile node;

(d) modifying the environment model according to said node locating step (c);

(e) updating the fingerprint according to the modification of the environment model in step (d); and

(f) repeating steps (c) to (e) in subsequent iterations.

The initial environmental model may be generated in step (a) by the system in response to the inputs or it may be received as a pre-generated file. The invention achieves by virtue of steps (b) to (5) self-calibration for location of mobile nodes in the environment. Any change made to the environment, such as modification of a base station, erection of a partition wall, or even movement of a piece of furniture will be automatically taken into account for further location of mobile nodes.

In one embodiment, the location engine is adapted to execute a filtering algorithm to estimate location in step (c).

In one embodiment, the processor is adapted to determine in step (c) possible mobile node trajectories, and to use said trajectories for environment model modification in step (d) and subsequent fingerprint updating in step (e).

In one embodiment, the processor is adapted to dynamically maintain a multi-hypothesis decision tree in step (c) in which it iteratively evolves branches representing the most likely mobile node trajectories.

In another embodiment, the processor is adapted to use mobile node speed estimation for steps (c) and (d).

In one embodiment, the processor is adapted to modify wall segment parameters of the environment model in step (d).

In one embodiment, in step (a) the inputs include:

layout of main obstacles including building floor plans and wall layout;
layout of base stations; and
starting position of a mobile node.

In a further embodiment, parameters of environment obstacles are initially set to a common default value.

In one embodiment, for the step (a) the processor is adapted to divide the obstacles into smaller segments and to optimise parameters of the smaller segments to more precisely represent each segment's mean local influence on signal attenuation.

In one embodiment, the processor is adapted to apply parameters to the segments so that they characterise the obstacles they are part of and also the influence of smaller obstacles including furniture in proximity to large obstacles.

In one embodiment, for step (c) the processor is adapted to use an empirical signal propagation model with a linear optimisation method for prediction of received signal strength.

In one embodiment, the processor is adapted to perform, for steps (c) and (d), simulation of an m^{th} mobile node motion along a h^{th} trajectory and creation of a corresponding fitness function.

In one embodiment, the fitness function describes how similar the channel parameters estimated along the h^{th} trajectory by the prediction model, which uses current obstacles parameters, is to gathered channel parameters by a mobile node along its truth trajectory.

In one embodiment, the fitness function takes the form of a set of linear equations for an optimisation method.

In one embodiment, the processor is adapted to execute a particle filter algorithm to locate a mobile node in step (c), in which particles are mapped to the physical environment as represented by the physical environment model.

In one embodiment, the processor is adapted to generate visual displays of the physical environment in which particles distribution is illustrated for user visualisation of likelihoods of locations of the mobile node.

In one embodiment, the processor is adapted to eliminate particles in successive iterations according to constraints imposed by the physical environment model and a motion model, in which remaining particles converge around the actual mobile node location.

In another embodiment, the processor is adapted to automatically recognise rooms in the physical environment model and to maintain a count of particles in each room to provide a per-room probability of presence of the mobile node.

In one embodiment, the processor is adapted to provide a random or uniform initial particle distribution mapped into the physical environment model, to filter the initial particle distribution with a likelihood observation function to generate a second particle distribution and to use said distribution for a next filtering iteration.

In one embodiment, the likelihood observation function is given by combining densities to obtain a likelihood observation function of measurement p(zt|xt), and is given by:

$$p(z_t|x_t) = \left[\prod_{k\in K} p_{hit}(z_t^k|x_t)\right]^{\frac{1}{|K|}} \cdot \prod_{k\in L} p_{miss}(z_t^k|x_t) \cdot \prod_{k\in M} p_{extra}(z_t^k|x_t) \cdot \beta \tag{3.27}$$

with

$$\beta = \frac{|K|}{|K| + |L| + |M|}$$

where:

K.................... is the set of measurement in ($z_t \cap z^+_t$),
L.................... is the set of measurement in ($z^+_t - z_t$),
M................... is the set of measurement in ($z_t - z^+_t$),

..............and the expressions of $|K|$, $|L|$, $|M|$ denotes the number of elements in K, L, M respectively,
.......................β represents the weighting factor and
.... k denotes AP identification.

In one embodiment, the processor is adapted to perform step (d) by adjusting the current parameters of obstacles towards the optimised parameters of obstacles which influence the channel parameters along the h^{th} trajectory, in which the adjustment is given by:

$$Lw^i_t = Lw^i_{t-1} + \Delta Lw^i_t$$,

where:

Lw^i_t............... represents i^{th} wall segment loss at time (index of iteration) t,
Lw^i_{t-1}............. represents i^{th} wall segment loss at iteration $t-1$, and
ΔLw^i_t............. denotes the delta of a wall segment loss at time t.

In another aspect, the invention provides a computer readable medium comprising software code adapted to perform operations of a location and tracking system as defined above in any embodiment when executing on a digital processor.

DETAILED DESCRIPTION OF THE INVENTION

Brief Description of the Drawings

The invention will be more clearly understood from the following description of some embodiments thereof, given by way of example only with reference to the accompanying drawings in which:-

Fig. 1 is a block diagram illustrating the main components of a location and tracking system of the invention, and Fig. 2 is a perspective view of a typical indoor environment in which the system operates;

Fig. 3 is a flow diagram illustrating operation of the system to perform mobile node tracking and self-calibration;

Fig. 4 is an example of a floor plan with a particular wall layout and access points, within which the location of a mobile node is tracked;

Fig. 5 is a screen shot for initializing the system with environment parameters;

Fig. 6 is another example floor plan, with segmented walls;

Fig. 7 shows possible trajectories in this environment, represented as a Voronoi diagram in the floor-plan;

Figs. 8 is display of fingerprints showing predicted channel parameters in an environment, in which the scale represents level of predicted channel parameter such as signal strength in a regular grid, and Fig. 9 shows another predicted fingerprint generated from a site-specific prediction model;

Fig. 10 shows a sample output screen for operation of a site survey module of the system, the screen showing signal strength as measured by a mobile node in the environment;

Fig. 11 is a set of diagrams illustrating proposed trajectories for a mobile node at a starting point A;

Fig. 12 is a diagram illustrating fitness function creation with a multi-wall model (MWM);

Fig. 13 is a plan view, showing a location error of the position estimated by a particle filter;

Fig. 14 is a diagram illustrating how the system dynamically maintains options and probabilities on possible trajectories;

Fig. 15 is a sample calibrated fingerprint generated by the system after several iterations of self-calibration, in which the fingerprint differs from the fingerprint shown in Fig 8 in the optimised parameters of the environment model (such as. wall parameters), which were used in the fingerprint predication mode; and

Figs. 16 to 23 are diagrams illustrating implementation of a particle filter algorithm.

Description of the Embodiments

Referring to Fig. 1 a location and tracking system 1 comprises a site survey tool 2, a location engine 3, a self-calibration engine 4, and a fingerprint database 5. The system operates for a wireless network of access points such as shown in Fig. 2. In hardware terms it comprises a conventional PC computer and mobile WiFi enabled handheld devices such as smart phones linked to wireless access points of a conventional WiFi network

The system 1 generates from initialization data a model of the physical environment. The model may for example be in the form of data representing geometry of lines in plan and signal attenuation values for the lines.

The system then uses this model to generate a fingerprint of projected signal strengths in the environment. The term “fingerprint” means a map of channel parameters (such as signal strength and Time of Arrival, TOA) at all of a set of points in the environment. These points may be in a regular grid pattern, with position of some points being offset to avoid being coincident with an obstruction such as a wall.

The system maintains a fingerprint for every environment model. There may be only one model at any one time, or there may be a number. Where the latter, the system may assign a weighting to each model to indicate its confidence value. For example if models are being maintained for a person moving through a building, the system may maintain a set of possible trajectories for the person. There might be a model for each trajectory, but a confidence weighting assigned according to the degree of confidence in the associated trajectory. In another example, the system is configured to maintain only one model at any time. This may correspond to only the trajectory for a single mobile node, and with the highest confidence value. The choice of number of models to maintain is dependent on the processing resources in the system and the extent of complexity of the environment, and the frequency with which it is changed.

The hardware of the system is an existing wireless communication network, and so no additional hardware is required where there is a wireless communication network such as a WiFi network in place.

Upon detection of a mobile node in the environment the system estimates position of the node according to the current fingerprint, and in real-time performs self-calibration of the environment model, in turn giving rise to fresh updated fingerprints for future tracking. This continuous self-calibration ensures excellent tracking accuracy even with modification of the environment, such as changing of an access point configuration, moving a partition wall, or movement of a metal filing cabinet.

The major steps, 101 – 122, implemented by the systems are shown in Fig. 3. The following describes these steps in detail.

<u>Step 101</u>

Input to the system 1 is:

- layout of main obstacles such as building floor plans or wall layout,
- layout of location and tracking system fixed nodes,
- initial (starting) position of mobile nodes (at the entrance to the building, for example).

Fig. 4 shows the example of a floor-plan with access points ("APs") and wall layout used as an input to the self-calibration system 1.

Step 102

Corresponding parameters of all environment obstacles and therefore all segments are initially set to the same default value. Default values are any reasonable mean values of obstacle parameters as characterised by the channel parameters prediction model used in the next point. The absolute values are not as important as their uniformity throughout the environment in the initial stage. A sample screen for inputting these parameters is shown in Fig. 5.

Step 103

The walls and main obstacles are then divided by the processor into smaller segments. The parameters of smaller segments can be better optimised to more precisely represent the segment's mean local influence on the signal attenuation than long walls. The segment's parameters therefore characterise not only the parameters of walls or main obstacles they are part of, but reflect also the influence of smaller surrounding obstacles such as furniture in the proximity of wall and variation of the longitudinal wall characteristic caused by pipes and ducting within the wall etc., which can have significant influence on the signal attenuation. The size of the segments depends on the type of environment and varies between 2 to 10 meters. We found that the segment size between 2 and 4 meters is suitable for indoor and between 5 and 10 meters for outdoor.

Algorithm 1 shows the algorithm for segmenting walls. The input of the algorithm is segment size (*l*), and output of the algorithm is a set consisting of segmented wall

Algorithm 1: *SegmentWall* (*l, W*)

```
for each wall in the floor plan
    $d$ = length of wall
    walls = div ($d, l$)
    if walls quotient > 1
        $w$ = segment the wall
        $W = W + w$
    else
        $w$ = the wall
        $W = W + w$
    end if
end for
```

```
10: return $W$
```

Fig. 6 shows example of an environment with segmented walls.

Step 104

The possible trajectories of mobile nodes in the described environment are pre-processed to accelerate the creation of a subset of possible trajectories in step 110. The trajectories are built as a Voronoi diagram of the floor-plan, the graph edge representing all the most likely routes for mobile node movement. The density of the Voronoi diagram varies depending on the predefined requirements on the location accuracy.

Algorithm 2 below shows the steps for creation of possible trajectories. The input of the algorithm is a vertex v in the Voronoi graph $\mathbf{G}$ and the distance travelled d by a mobile device. The output of the algorithm is a set of vertices $\mathbf{V}$, which is basically a set of way-points inside the trajectory

```
Algorithm 2: Traverse($v$, $\mathbf{V}$, $d$)
1: $\mathbf{V} = \mathbf{V} + \{v\}$
2: $D$ = total length of edges in $\mathbf{V}$
3: if $D < d$ then
4: for each vertex $w$ adjacent to $v$ do
5: Traverse( $w$, $\mathbf{V}$, $d$)
6: end for
7: else
8: save $\mathbf{V}$ {$\mathbf{V}$ is a candidate of trajectory}
9: end if
```

Fig. 7 shows a screenshot of the possible subset of motion trajectories in a floor-plan.

Step 105

Estimation of the expected channel parameters fingerprint in the area of interest taking into account the default parameters of obstacles by appropriate empirical or physical channel parameters prediction models.

In this implementation a simple empirical signal propagation model, called Multi-Wall-Model (MWM), is used for prediction of received signal strength (RSS) fingerprint. MWM is chosen because of its simplicity and efficient computation, but also for its suitability to the linear optimisation method used in this implementation. Both of these factors contribute to improving the speed of the overall self-calibration.

In the MWM, the path loss is given by the following equation:

$$L = L_0 + 20\log(d) + \sum_{i=1}^{I} Lw^i + L_f \tag{1}$$

Where:

L .... denotes the predicted signal loss,

L_0 .... is the free space loss at a distance of 1 meter from the transmitter,

d .... is the distance from the transmitter to the receiver, .

I .... is the number of walls intersecting with the signal direct propagation path between the transmitter and receiver (access point and mobile node),

Lw^i .... denotes the contribution of i^{th} wall segment to the total signal loss,

L_f .... is the contribution of floors to the total signal loss.

In this example, self-calibration is used for localisation on a single floor, therefore signal-loss of floors L_f is not considered.

Algorithm 3 below shows the algorithm for signal level prediction with the multi-wall model. The input of the algorithm is position of an access point p. The output is predicted signal level in the environment S.

Algorithm 3: *Prediction* (p)

```
1: $P$ = -10 dB
2: S = Ø
3: for each grid $g$ in the environment
4: $d$ = distance from $g$ to access point $p$
5: $L$ = calculate with equation (1)
6: $s = P - L$
```

```
    $S = S + s$
end for
return $S$
```

Step 106

Start of the main loop of an iterative fingerprint self-calibration algorithm by the engine 4.

Step 107

Mobile nodes, subject to their pinpointing by the location engine 3, gather the observed channel parameters along their motion trajectory, and send them to the survey tool 3. This interval of gathering of parameters can vary from a few seconds to many minutes based on contextual or other information. A sample screen is shown in Fig. 10. The plots at the bottom of this screen are signal strength from different transceivers as seen by the mobile node.

Step 108

The gathered channel parameters could be optionally used by the location engine 3 to estimate motion behaviour of the mobile nodes such as speed and direction.

The motion behaviour estimation can be further improved by additional sensors, especially inertial sensors such as accelerometers, gyroscopes and magnetometers built-in the mobile node. The suitable methods used for estimating the mobile node speed and directions based on measurement of the inertial sensors are called "Dead Reckoning" methods.

Step 109

This is the start of a loop through all participating mobile nodes, *M*, by the engine 4.

Step 110

Based on the length of the time interval of gathering of channel parameters (also an option on estimated mobile node motion behaviour) and the layout of obstacles in the environment, possible H_m motion trajectories (hypothesis) of the m^{th} mobile node are proposed.

Fig. 11 shows an example of proposed trajectories when a mobile node is at the Starting point A

Step 111

This is start of an inner loop for evaluating the likelihood (ranking) of all proposed trajectories (hypothesis), H_m.

Step 112

This includes simulation of the m^{th} mobile node motion along the h^{th} trajectory and creation of the corresponding fitness function. The fitness function describes how similar the channel parameters estimated along the h^{th} trajectory by the prediction model, which uses current obstacle parameters, is to the gathered channel parameters by a mobile node along its truth trajectory. The fitness function can take the form of a set of linear equations suited to the optimisation method used in the following step.

In this implementation the localisation is performed on a single floor. Therefore the fitness function for the optimisation of a wall segment parameter considering MWM (1) is written as follows:

$$\begin{aligned} L &= L_1 + 20\,log(d) + Lw \\ P^k - z_t^k &= L_1 + 20\,log(d) + Lw_{t-1} + \Delta Lw_t \\ \Delta Lw_t &= P^k - z_t^k - L_1 - 20\,log(d) - Lw_{t-1} \end{aligned} \tag{2}$$

Where:

$$Lw = \sum_{i=1}^{I} Lw^i, \quad \Delta Lw = \sum_{i=1}^{I} \Delta Lw^i$$

ΔLw .... denotes the difference of wall segment loss,

P^k .... denotes the power transmitted from access point k, (it is known or can be estimated as additional unknown similarly to by the described optimisation method as if it was an additional wall around the transmitter)

$z^k{}_t$ .... represents the RSS measurement gathered by the mobile device from access point k,

t...................... denotes time or index of process iteration

Lw^i .... represents an i^{th} wall segment loss and

ΔLw^i.............. denotes the difference of an i^{th} wall segment loss.

Linear equations (2) are then created for every measurement point along the simulated h^{th} trajectory. A created set of the linear equations represents all wall segments which have influence on the signal strength along the trajectory. Since the number of measurements is usually greater than the number of wall segments, the resulting set of linear equations form an over-determined system. This means that the number of equations is greater than the unknowns (i.e: ΔLw). A fast and robust algorithm to solve an over-determined linear system is the least-squares method.

Considering this over determined system where m represents the number of linear equations and n is the number of unknown coefficients with $m > n$. Recalling the fitness function in equation (2), β is the ΔLw, and y_i represents the right hand side of the equation.

The linear system is written:

$$\sum_{j=1}^{n} X_{ij}\beta_j = y_i (i = 1, 2, \ldots, m)$$

and the final fitness function in the matrix form suitable for example to the least-squares method:

$$\mathbf{X}\beta = \mathbf{y} \tag{3}$$

Where:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, \beta = \begin{bmatrix} \Delta Lw_t^1 \\ \Delta Lw_t^2 \\ \vdots \\ \Delta Lw_t^m \end{bmatrix} \text{ and}$$

$$\mathbf{y} = \begin{bmatrix} P^1 - z_t^1 - L_1 - 20\,log(d^1) - Lw_{t-1}^1 \\ P^2 - z_t^2 - L_1 - 20\,log(d^2) - Lw_{t-1}^2 \\ \vdots \\ P^m - z_t^m - L_1 - 20\,log(d^m) - Lw_{t-1}^m \end{bmatrix}$$

<u>Step 113</u>

This step involves application of a formal optimisation method (such as the least square, agent-based, or stochastic search methods) to find optimal parameters of obstacles which maximise the similarity of the predicted channel parameters and gathered channel parameters along the h^{th} trajectory.

In this implementation the least-squares method is used to solve the matrix (3), where the solution is given by:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

Step 114

This is adjustment of the current parameters of obstacles towards the optimised parameters of obstacles which influence the channel parameters along the h^{th} trajectory.

The adjustment is given by:

$$Lw_t^i = Lw_{t-1}^i + \Delta Lw_t^i$$,

where:

Lw^i_t ....represents i^{th} wall segment loss at time (index of iteration) t,

Lw^i_{t-1} ....represents i^{th} wall segment loss at iteration $t-1$,

ΔLw^i_t..............denotes the delta of a wall segment loss at time t.

Step 115

This step involves prediction (re-calculation) of the channel parameters fingerprint in the area of interest with a prediction model having adjusted parameters of obstacles.

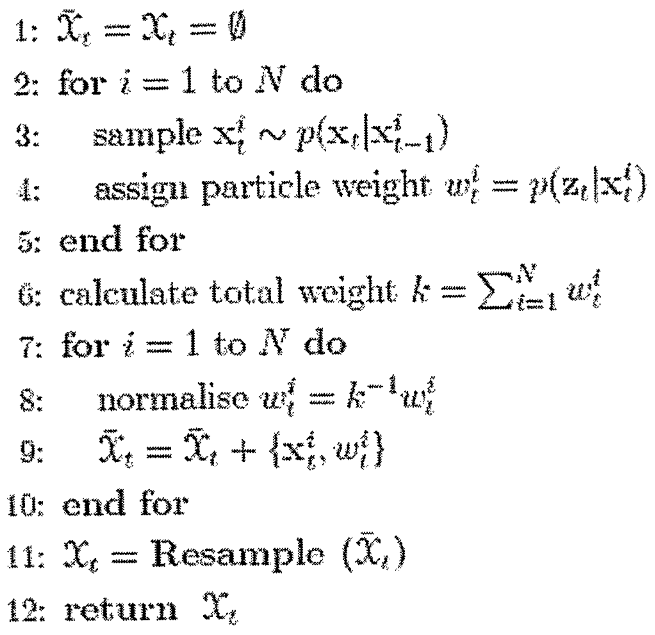
Step 116

This step involves location error estimation, which takes into account the re-calculated fingerprint and gathered channel parameters and the h^{th} trajectory as a ground truth.

Fig. 13 shows an illustration of the localization error.

In this implementation the Particle Filter algorithm (Algorithm 4) is used to estimate the location error, as follows.

Algorithm 4: *Particle Filter* (χ_{t-1})



```
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $i = 1$ to $N$ do
    sample $\mathrm{x}_t^i \sim p(\mathrm{x}_t|\mathrm{x}_{t-1}^i)$
    assign particle weight $w_t^i = p(\mathrm{z}_t|\mathrm{x}_t^i)$
end for
calculate total weight $k = \sum_{i=1}^{N} w_t^i$
for $i = 1$ to $N$ do
    normalise $w_t^i = k^{-1} w_t^i$
    $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \{\mathrm{x}_t^i, w_t^i\}$
end for
$\mathcal{X}_t$ = Resample $(\bar{\mathcal{X}}_t)$
return $\mathcal{X}_t$
```

where:

$\bar{\chi}_t, \chi_t$ ....is the set of particles,

N ....is the number of particles,

x_t^i ....is a particle state including particle location,

w_t ....is a particle weight,

$p(x \mid x_t^i)$ ....is transition distribution and

$p(z \mid x_t^i)$ ....is measurement distribution.

Implementation of step 116 is described in more detail in a separate section below.

<u>Step 117</u>

The actual ranking (the likelihood estimation) of the h^{th} trajectory is calculated taking primarily into account the localisation error and also other contextual information such as the motion behaviour pattern of the mobile nodes in the area of interest.

In the presented implementation the h^{th} trajectory likelihood is calculated as a probability govern by Gaussian distribution with the mean localisation error of 0 meters and standard deviation of 3m by the following expression:

$$\omega^h = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) \qquad (5)$$

where:

ω^h..................denotes trajectory likelihood,

σ.................represents the standard deviation of the Gaussian distribution.

ε ....represents the localisation error and

Localisation error in (5) is calculated as the mean Euclidian distance between the mobile node locations estimated by a formal localisation algorithm such as the particle filter and the ground truth positions along the h^{th} trajectory

$$\varepsilon = \frac{1}{N} \sum_N \left\| \hat{x} - x_{GT} \right\| \qquad (6)$$

where:

N...................is the number of the measurement along the h^{th} trajectory,

x_{GT} ....are the ground truth positions and

$\hat{x}$ ....are the positions estimated by a formal localisation algorithm.

Step 118

The end of the trajectories (hypothesis) likelihood evaluation loop.

Step 119

Selection of a limited number of the most likely trajectories (hypothesis) for the h^{th} mobile node. The other trajectories are terminated.

Fig. 14 shows an example of the evolution of the self-calibration process as a hypothesis tree for a mobile node where individual branches represent its possible trajectories, which are maintained or terminated depending on their estimated likelihood. The most likely trajectory is the trunk of the tree.

The system configuration in this implementation sets the threshold for the trajectories termination to 0.05. The threshold value impacts the number of maintained hypotheses, therefore, the computational requirements of the self-calibration system.

Step 120

This is the end of the mobile node multi-hypothesis trajectory evaluation. At the end of this loop every participating mobile node has an associated set of most likely hypotheses of trajectories.

Step 121

The process of ranking combined hypothesis follows the same steps as the ranking of the individual node trajectories from points 9 to 16, however, the optimisation of parameters of obstacles can be simplified by averaging, for example.

Step 122

This is selection of a limited number of the most likely combined-trajectories, H_c. The other trajectories (hypotheses) are discontinued.

Fig. 15 shows a final predicted fingerprint based on the calibrated wall segment parameters. The scale represents level of predicted channel parameter such as signal strength in regular grid

Particle Filter Algorithm: Step 116

The particle filter is a non-parametric implementation of the Bayes' filter. It approximates the posterior probability by a finite number of discrete samples with associated weights, called particles. The approximation of the posterior density is non-parametric. Therefore, it can represent a wider distribution than the parametric one, such as Gaussian. The particles are mapped to the physical environment model, and so the processor can generate visualisation user displays of the environment with the particle distribution illustrated.

The particle filter directly estimates the posterior probability of the state expressed in the following equation:

$$p(\mathrm{x}_t|z_t) \approx \sum_{i=1}^{N} w_t \delta(\mathrm{x}_t - \mathrm{x}_t^i) \tag{7}$$

where:

x^i_t............is the i^{th} sampling point or particle of the posterior probability with $1 < i < N$,

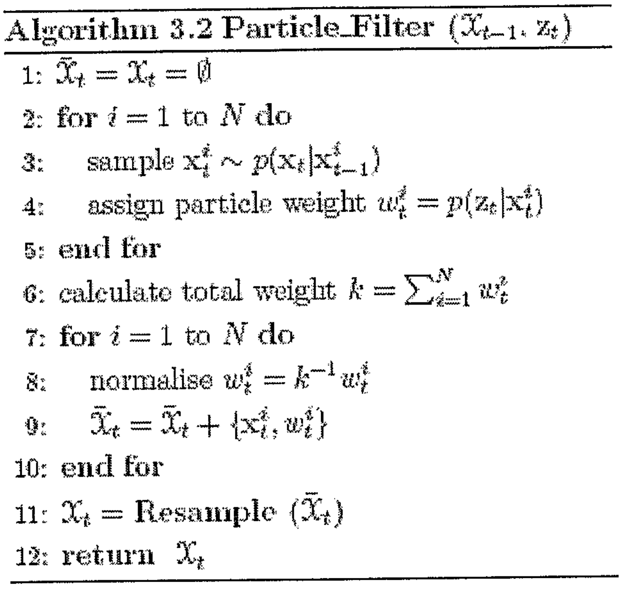
w^i_t............is the weight of the particle and

N..............represents the number of particles in the particle set, denoted by X_t.

$$\mathcal{X}_t := \mathrm{x}_t^1, \mathrm{x}_t^2, \mathrm{x}_t^3, \ldots, \mathrm{x}_t^N \tag{8}$$

Each particle is a concrete instantiation of the state at time t, or put differently, each particle is a hypothesis of what the true state x_t might be, with a probability given by its weight.

The property of equation (eqn. 7) holds for $N \uparrow \infty$. In the case of finite N, the particles are sampled from a slightly different distribution. However, the difference is negligible as long as the number of particles is not too small.



Algorithm 3.2 Particle_Filter ($\mathcal{X}_{t-1}, \mathrm{z}_t$)

```
1: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
2: for $i = 1$ to $N$ do
3:     sample $\mathrm{x}_t^i \sim p(\mathrm{x}_t | \mathrm{x}_{t-1}^i)$
4:     assign particle weight $w_t^i = p(\mathrm{z}_t | \mathrm{x}_t^i)$
5: end for
6: calculate total weight $k = \sum_{i=1}^{N} w_t^i$
7: for $i = 1$ to $N$ do
8:     normalise $w_t^i = k^{-1} w_t^i$
9:     $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \{\mathrm{x}_t^i, w_t^i\}$
10: end for
11: $\mathcal{X}_t$ = Resample ($\bar{\mathcal{X}}_t$)
12: return $\mathcal{X}_t$
```

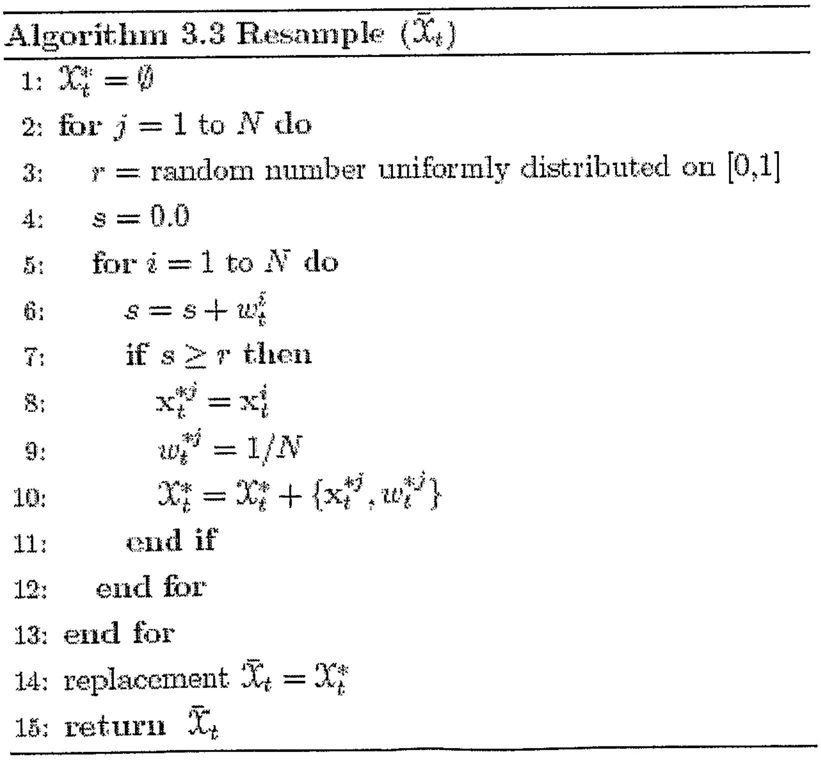
The algorithm 3.2 above describes a generic particle filter algorithm. The input of the algorithm is the previous set of the particle X_{t-1}, and the current measurement z_t, whereas the output is the recent particle set X_t. The algorithm will process every particle x^i_{t-1} from the input particle X_{t-1} as follows.

Line 3 shows the prediction stage of the filter. The particle x^i_{t} is sampled from the transition distribution $p(x_t \mid x_{t-1})$. The set of particles resulting from this step has a distribution according to (denoted by ~) the prior probability $p(x_t \mid z_{t-1})$. Line 4 describes incorporation of the measurement z_t into the particle. It calculates for each particle x^i_t the importance factor or weight w^i_t. The weight is the probability of particle x^i_t received measurement z_t or $p(z_t \mid x_t)$. Lines 7 to 10 are steps required to normalise the weight of the particles. The result is the set of particles $\bar{\mathcal{X}}_t$, which is an approximation of posterior distribution $p(x_t \mid z_t)$. Line 11 describes the step which is known as re-sampling or importance re-sampling. After the re-sampling step, the particle set, which was previously distributed in a manner equivalent to prior distribution $p(x_t \mid z_{t-1})$ will now be changed to particle set X_t which is distributed in proportion to $p(x_t \mid z_t)$.

Re-sampling

The early implementation of particle filter is called sequential important sampling (SIS). Implementation of SIS is similar to algorithm 3.2 above, but without the resample step (algorithm 3.2 line 11). The SIS approach suffers an effect which is called the *degeneracy problem*. This occurs when, after some sequence time *t*, all but one particle has nearly zero weight. The problem happen since generally impossible to sample directly from posterior distribution, particles are sampled from related distribution, termed importance sampling. The choice of importance sampling made the degeneracy problem is unavoidable. The degeneracy problem will increase the weight variance over time and has a harmful effect on accuracy.

Re-sampling is an important step to overcome the degeneracy problem. It will force the particle to distribute according to the posterior density $p(x_t \mid z_t)$.



```
Algorithm 3.3 Resample ($\bar{\mathcal{X}}_t$)
$\mathcal{X}_t^* = \emptyset$
for $j = 1$ to $N$ do
    $r$ = random number uniformly distributed on [0,1]
    $s = 0.0$
    for $i = 1$ to $N$ do
        $s = s + w_t^i$
        if $s \geq r$ then
            $\mathrm{x}_t^{*j} = \mathrm{x}_t^i$
            $w_t^{*j} = 1/N$
            $\mathcal{X}_t^* = \mathcal{X}_t^* + \{\mathrm{x}_t^{*j}, w_t^{*j}\}$
        end if
    end for
end for
replacement $\bar{\mathcal{X}}_t = \mathcal{X}_t^*$
return $\bar{\mathcal{X}}_t$
```

Algorithm 3.3 above describes a re-sampling step of particles. The modification lies in the iteration to draw newly sampled particles. The input of the algorithm is a set of particles and the output is the new resampled particles.

Line 3 shows the generation of random number r. Lines 5 to 8 show the re-sampling steps. Line 8 shows when a particle is re-sampled only if its weight contribution causes the summation in line 6 to exceed or equal to the random sample r. The probability of drawing a particle is in proportion to its importance factor. Particles with lower weight will have a lower chance and some will not be resampled. Since the re-sampling process is performed every time step, the new particles do not need to know the old weight and their weight will be set equally, as shown in line 9. Line 14 shows when members of particle set $\bar{\mathcal{X}}_t$ is replaced by new resampled particles.

The particle filter, which is a non-parametric implementation of Bayes' filter, is well suited for state estimation in opportunistic system for a number of reasons. It is suitable for state estimation for a system that has large process noise in state dynamics, and it provides a good processor performance.

To implement a particle filter for opportunistic indoor localisation, motion and measurement probabilistic models have been devised, s described below..

Motion Model

This implements a particle filter for and opportunistic system. It is a representation of the target's kinematics behaviour. It is used to construct the transition probability $p(x_t \mid x_{t-1})$ which has an important role for the prediction step in the particle filter. The first-order of person kinematics is less conservative than Brownian movement, in a way that the motion is not totally random as in Brownian movement. Therefore it is often more accurate. It assumes that the target direction is the same as the last observing movement. However, this assumption does not hold when people turn direction in a corner, enter a room, or avoid obstacles. This kind of action therefore is difficult to represent. Our approach for the motion model in indoor localisation is taken from both Brownian movement and a first-order motion model. The motion model assumes that the kinematics of a target has random values in it. However, this randomness is constrained by the previous state. This approach accurately reflects the target's kinematics behaviour (i.e.: a person) in opportunistic system. Especially, in a system that only uses RSS measurement, without additional motion sensor.

Velocity

To accommodate simple human locomotion patterns (such as remaining stationary, walking and running) and its limitations, the target velocity is constrained to take place only between limited ranges of speed. The succeeding velocity also takes into account the preceding one, since it is assumed that people tend not to change their kinematics abruptly in a normal condition. Based on observation of peoples' movement in the standard environment, the target velocity is given by the following rule:

$$v_t = \mathcal{N}(v_{t-1}, \sigma_v) \text{ and } v_t = \begin{cases} v_t, & 0 \leq v_t \leq Max_{v_t} \\ v_t = |v_t|, & v_t < 0 \\ 2\,Max_{v_t} - v_t, & Max_{v_t} < v_t \end{cases} \tag{9}$$

with

$$\sigma_v = \min(Max_{\Delta_t}, \sqrt{\Delta_t}) \tag{10}$$

where:

N .... represents a Gaussian random number generator,

Max_{vt} .... represents the maximum speed.

min() .... is a function which returns the smallest component,

Δ_t .... represents elapsed time,

$\text{Max}_{\Delta t}$ .... represents the maximum threshold of the Δt.

The value of Max_{vt} is set to the fastest recorded human speed (≈ 10 m/s) and $Max_{\Delta t}$ = 3s. Fig. 16 shows an example of the velocity probability distribution function (PDF) when v_{t-1} = 0 m/s with various Δ_t.

Direction

The succeeding direction also considers that preceding it. A velocity parameter is also included in the direction calculation. The motivation is to limit heading variation based on preceding speed, since people tend to slow down when they want to change their direction of movement. The target direction is given by the following rule:

$$\alpha_t = \mathcal{N}(\alpha_{t-1}, \sigma_h) \text{ and } \alpha_t = \begin{cases} \alpha_t, & -\pi \leq \alpha_t \leq \pi \\ \alpha_t + 2\pi, & \alpha_t < -\pi \\ \alpha_t - 2\pi, & \alpha_t > \pi \end{cases} \tag{11}$$

with

$$\sigma_h = 0.4\pi - \arctan(\frac{\sqrt{v_{t-1}}}{2}) \qquad (12)$$

where α_t represents direction at time t, σ_h represents the direction standard deviation.

Fig. 17 shows the σ_h for different velocities. Figure 18 presents PDF of the direction based on different velocities with $\alpha_{t-1} = 0$. It can be seen that the heading variation is narrower when the velocity is increased.

Particle Motion Model

Each particle, which is a hypothesis of the target state in the real world, will have a kinematic characteristic according to the motion model during the prediction stage. The particle state can be modelled with:

$$\mathbf{x}_t^i = \begin{bmatrix} x_t^i \\ y_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i + v_t^i cos(\alpha_t^i)\Delta t + n_t \\ y_{t-1}^i + v_t^i sin(\alpha_t^i)\Delta t + n_t \end{bmatrix} \qquad (13)$$

where :

v_t .... denotes velocity,

α_t .... describes particle direction at the time t,

n_t .... is a noise with Gaussian distribution.

The particles' velocity is governed by equation (9), whereas particles' direction is given by equation (11). The noise n_t is added to achieve a better distribution by preventing particles from collapsing into a single point. A n_t value of 0.5m is used for this work.

Fig. 19 illustrates the evolution of particle distribution determined only by the motion model. The particle distribution is started from a known state in 2D space. The initial velocity $v_0 = 3$ m/s and number of particles are 2000. Since there is no RSS measurement update performed, the distribution spreads wider over time.

The aforementioned particle distribution, which was constructed with the motion model, is a representation of transition probability. This density needs to be updated with the measurement probability $p(z_t \mid x_t)$ to obtain the posterior distribution. The construction of measurement probability with the measurement model in an opportunistic system is explained in the next section.

Measurement Model

The measurement model describes a construction process by which the sensor measurement is generated. This model is utilised to obtain the measurement probability of $p(z_t \mid x_t)$, often referred to as the likelihood observation function. In the particle filter, this function is used to incorporate a measurement update into the particle weight w^i_t.

There is a measurement model which considers the physical characteristics of signal propagation in indoor environments based on the fingerprinting method.

A fingerprint is a database of stored RSS measurements throughout the coverage area of all discrete possible states. These discrete states are often referred to as a grid. The fingerprint acts as the "true" RSS values obtained by the sensor, denoted by z^+_t , at the state x_t. Theoretically, smaller grid size gives better localisation accuracy since the measurement model will be more distinct between grids. However, it is technology dependent. Based on the observations, WLAN RSS values are discernible when they are measured in a grid ≥ 1m, for example.

To obtain the likelihood observation function, statistical inference is performed between a true value stored in the fingerprint and recent RSS measurement (termed as *signature*).

The mobile device is able to simultaneously retrieve RSS measurements from different APs, therefore z_t is a set of measurements which is described as:

$$z_t = \{z_t^1, z_t^2, z_t^3, \ldots, z_t^K\} \qquad (14)$$

where:

z^k_t .... denotes RSS value z_t at time t from AP with identification k (MAC address is used for WLAN AP)

The probability $p(z_t \mid x_t)$ is obtained as the product of the individual likelihood observation function.

$$p(z_t|x_t) = \prod_{k \in z_t} p(z_t^k|x_t) \qquad (15)$$

Algorithm for the Measurement Model

The measurement model comprises three types of probability densities, each of which corresponds to a type of dissimilarity:

1. Measurement with noise. Assuming that the sensor can capture the true RSS measurement, the returned value is still subject to error caused by shadowing (or slow fading) and multipath fading. This fading, which appears as a time-varying process, makes the RSS value fluctuate even when it is measured at the same position.

   Since the sensor (i.e. WLAN interface) has an internal mechanism to suppress multipath fading, the received error is largely caused by shadowing which has a Gaussian distribution. Therefore, the noisy measurement may be modelled with a Gaussian distribution with mean value of z^{+k}_{t} and standard deviation of σ^{+}. The Gaussian is denoted by p_{hit} as described in Figure 20.

   The value of z^{+k}_{t} is given by the RSS measurement stored in the corresponding grid of the fingerprint. The measurement probability is governed by

$$\begin{aligned} p_{hit}(z_t^k|x_t) &= \mathcal{N}(z_t^k; z_t^{+k}, \sigma_+) \\ &= \frac{1}{\sigma_+\sqrt{2\pi}} \exp\left(-\frac{(z_t^k - z_t^{+k})^2}{2\sigma_+^2}\right) \end{aligned} \qquad (16), (17)$$

   where z^{+k}_{t} is the RSS value stored in the fingerprint and z^{k}_{t} is the recent signature such that $\{k \in z_t \cap z^{+}_{t} \}$ or, to put it differently, the RSS value from an AP which appears in both signature and fingerprint. Notation $\sigma+$ is the intrinsic noise parameter for the measurement model.

2. Missing RSS. Considering that it may be the case that not all of the RSS values that appear in the fingerprint grid and signature coming from the same APs, a question arises as to how to account for such data. This condition can happen when the signature is taken at a different location or there is an extra/missing AP. In the case of no extra/missing AP, a different set of RSS values in the fingerprint grid and signature means that they are taken at different locations. Incorporating this information into the measurement model will eventually lead to better localisation accuracy.

In the case of RSS measurements from an AP which is missing in the signature but which is present in the fingerprint, the measurement probability is modelled by a Gaussian denoted by p_{miss}. The measurement probability is governed by:

$$\begin{aligned} p_{miss}(z_t^k|\mathrm{x}_t) &= \mathcal{N}(\eta;0,\sigma_+) \\ &= \frac{1}{\sigma_+\sqrt{2\pi}}\exp\left(-\frac{\eta^2}{2\sigma_+^2}\right) \end{aligned} \qquad (18), (19)$$

where η is the penalty parameter given for the missing RSS. The density p_{miss} is calculated for missing z^k_t such that $\{k \in z^+_t - z_t\}$.

3. Extra RSS. In the case where there is an extra RSS value of an AP which exists in the signature but which is absent in the fingerprint, the measurement probability is also modelled by a Gaussian denoted by p_{extra} and governed by:

$$\begin{aligned} p_{extra}(z_t^k|\mathrm{x}_t) &= \mathcal{N}(\eta;0,\sigma_+) \\ &= \frac{1}{\sigma_+\sqrt{2\pi}}\exp\left(-\frac{\eta^2}{2\sigma_+^2}\right) \end{aligned} \qquad (20), (21)$$

where η is the penalty parameter given for the extra RSS. The density p_{extra} is calculated for extra z^k_t such that {k $\in z_t - z^+_t$ }.

These three densities are combined to obtain the likelihood observation function of RSS measurement p(zt|xt). The likelihood observation function is given by:

$$p(\mathrm{z}_t|\mathrm{x}_t) = \left[\prod_{k\in K} p_{hit}(z_t^k|\mathrm{x}_t)\right]^{\frac{1}{|K|}} \cdot \prod_{k\in L} p_{miss}(z_t^k|\mathrm{x}_t) \cdot \prod_{k\in M} p_{extra}(z_t^k|\mathrm{x}_t) \cdot \beta \qquad (22)$$

with

$$\beta = \frac{|K|}{|K|+|L|+|M|}$$

where:

K.................... is the set of measurement in $(z_t \cap z^+_t)$,

L.................... is the set of measurement in $(z^+_t - z_t)$,

M.................. is the set of measurement in $(z_t - z^+_t)$.

The expressions of $|K|$, $|L|$, $|M|$ denotes the number of elements in K, L, M respectively.

β.................... represents the weighting factor and

k..................... denotes AP identification.

Algorithm 3.4 describes the steps necessary to obtain the likelihood observation function. The input of the algorithm is the measurement z_t and the state x_t. The output is the likelihood observation function of $p(z_t \mid x_t)$.

Algorithm 3.4 Measurement_Model $(\mathrm{z}_t, \mathrm{x}_t)$

```
$p = 1$
$q = 1$
$y = 1$
$S = 0$
get $\mathrm{z}_t^+$ from the fingerprint
for $k \in K$ do
    $p = p.p_{hit}(z_t^k|\mathrm{x}_t)$
end for
$p = p^{1/|K|}$
for $k \in L$ do
    $q = q.p_{miss}(z_t^k|\mathrm{x}_t)$
end for
for $k \in M$ do
    $q = q.p_{extra}(z_t^m|\mathrm{x}_t)$
end for
$S = |K| \div (|K| + |L| + |M|)$
$y = p.q.S$
return $y$
```

The intrinsic noise parameter for the measurement model $\sigma+$ is set to 4dBm which was observed as a appropriate value for common indoor environments.

The value of the penalty η in equation (19) of p_{miss} and (21) of p_{extra} is governed by the following expression

$$\eta = \begin{cases} (5/30)(z_t + 100), & z_t \leq -70 \\ (15/25)(z_t + 70) + 5, & -70 < z_t \leq -45 \\ 20, & z_t > -45 \end{cases} \tag{23}$$

The aforementioned equation is formulated to mimic the tail part of a Gaussian distribution. The RSS range is taken from maximum and minimum RSS values measurable from a WLAN interface (-100dBm and -45dBm). Maximum penalty is given by the maximum signal loss caused by people shadowing (20dBm). Linear interpolation is used to obtained penalty η values.

The result is that the value of η is in proportion to the RSS value missing in the fingerprint z^{+}_{t} or extra in z_t. Greater missing / extra RSS values. will have a proportionately higher penalty η (Fig. 21).

The value of intrinsic parameters of standard deviation σ_{+} and penalty η can be applied to environment similar with office setting (characterised by many walls, corridors and rooms), such as hospital, campus or apartment.

Examples

Fig. 22 is used to illustrate how to obtain the likelihood observation function of a particle $p(z_t \mid x^{1}_{t})$. It shows a fingerprint from 3 APs, a target and a particle.

Supposing that the target measured RSS value z_t = {AP1 = -70dBm, AP2 = -70dBm, AP3 = -75dBm}. The corresponding fingerprint grid where the particle x is located, has value of z^{+}_{t} = {AP3 = -65dBm}. If the value of both σ_{+} and η are set to 4 dBm, the $p(z_t \mid x^{1}_{t})$ can be calculated as follows:

Calculating p_{hit} from AP3 with equation (19):

$$\begin{aligned} p_{hit}(z_t^3|\mathrm{x}_t^1) &= \frac{1}{\sigma_+\sqrt{2\pi}}\exp\left(-\frac{(z_t^k - z_t^{+k})^2}{2\sigma_+^2}\right) \\ &= \frac{1}{4\sqrt{2\pi}}\exp\left(-\frac{(-75-(-65))^2}{2.4^2}\right) = 0.0044 \end{aligned}$$

Calculating p_{extra} from AP1 and AP2 with equation (21):

$$\begin{aligned} p_{extra}(z_t^1|\mathrm{x}_t^1).p_{extra}(z_t^2|\mathrm{x}_t^1) &= \left(\frac{1}{\sigma_+\sqrt{2\pi}}\exp\left(-\frac{\eta^2}{2\sigma_+^2}\right)\right)^2 \\ &= \left(\frac{1}{4\sqrt{2\pi}}\exp\left(-\frac{(4)^2}{2.4^2}\right)\right)^2 = 0.0037 \end{aligned}$$

Calculating $p(z_t \mid x^{1}_{t})$ with equation (22):

$$p(z_t|\mathbf{x}_t^1) = 0.0044\,.\,0.0037\,.\,\frac{1}{1+2} = 0.0000054$$

Measurement Model & Particles Distribution

The likelihood observation function update to the particle distribution is illustrated in Fig. 23. Fig. 23(a) shows three access points and its RSS measurement throughout the coverage area which is used as the fingerprint. In the initial time, the particles will be uniformly distributed as shown in Fig. 23(b). Once measurement z_t is obtained, the likelihood observation function or measurement probability is calculated based on equation (22). The result is illustrated in Fig. 23(c). Higher likelihoods are shown with bigger blue circles. Furthermore, the particles' weight is updated with the likelihood observation function and subsequently is resampled. The new particle distribution and estimated target location are shown in Fig. 23(d).

Estimation of Target State

Once the posterior probability is obtained, the state is estimated using a particle set X_t which is distributed in proportion to $p(x_t\,|\,z_t)$ (X_t is a return value of algorithm 3.2 of particle filter). Algorithm 3.5 below describes steps to estimate this state.

```
Algorithm 3.5 State_Estimation($\mathcal{X}_t$)
$\hat{\mathbf{x}}_t = 0$
for $i = 1$ to $|\mathcal{X}_t|$ do
    $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t + \mathbf{x}_t^i . w_t^i$
end for
return $\hat{\mathbf{x}}_t$
```

It will be appreciated that the invention enables an automatic generation and update of the channel parameters fingerprint, merely initially requiring knowledge of layout of dominant obstacles and system infrastructure in the environment (the floor plan and system fixed node positions, for example). This is not onerous. It will also be appreciated that the invention avoids need for dedicated hardware, as it can use the hardware of an existing wireless network such as a WiFi network.

The invention is not limited to the embodiments described but may be varied in construction and detail. For example, it may be applied to outdoor location and tracking systems in particular areas. Also, the initial physical environmental model may be received in the initial inputs rather then being generated by the system.

Claims

1. A location and tracking system comprising wireless base stations, an input interface, and a processor, wherein the processor is adapted to

(a) receive physical environment inputs and provide an initial physical environment model;

(b) generate an initial channel parameters fingerprint for the environment according to the initial environment model;

(c) monitor channel parameters of a mobile node in the environment and use said parameters and the fingerprint to locate the mobile node;

(d) modify the environment model according to said node locating step (c);

(e) update the fingerprint according to the modification of the environment model in step (d); and

(f) repeat steps (c) to (e) in subsequent iterations.

2. A location and tracking system as claimed in any preceding claim, wherein the location engine is adapted to execute a filtering algorithm to estimate location in step (c).

3. A location and tracking system as claimed in claims 1 or 2, wherein the processor is adapted to determine in step (c) possible mobile node trajectories, and to use said trajectories for environment model modification in step (d) and subsequent fingerprint updating in step (e).

4. A location and tracking system as claimed in claim 3, wherein the processor is adapted to dynamically maintain a multi-hypothesis decision tree in step (c) in which it iteratively evolves branches representing the most likely mobile node trajectories.

5. A location and tracking system as claimed in claims 3 or 4, wherein the processor is adapted to use mobile node speed estimation for steps (c) and (d).

6. A location and tracking system as claimed in any preceding claim, wherein the processor is adapted to modify wall segment parameters of the environment model in step (d).

7. A location and tracking system as claimed in any preceding claim, wherein in step (a) the inputs include:

   layout of main obstacles including building floor plans and wall layout;

   layout of base stations; and

   starting position of a mobile node.

8. A location and tracking system as claimed in any preceding claim, wherein parameters of environment obstacles are initially set to a common default value.

9. A location and tracking system as claimed in claims 7 or 8, wherein for the step (a) the processor is adapted to divide the obstacles into smaller segments and to optimise parameters of the smaller segments to more precisely represent each segment's mean local influence on signal attenuation.

10. A location and tracking system as claimed in claim 9, wherein the processor is adapted to apply parameters to the segments so that they characterise the obstacles they are part of and also the influence of smaller obstacles including furniture in proximity to large obstacles.

11. A location and tracking system as claimed in any preceding claim, wherein for step (c) the processor is adapted to use an empirical signal propagation model with a linear optimisation method for prediction of received signal strength.

12. A location and tracking system as claimed in any of claims 3 to 11, wherein the processor is adapted to perform, for steps (c) and (d), simulation of an m^{th} mobile node motion along a h^{th} trajectory and creation of a corresponding fitness function.

13. A location and tracking system as claimed in claim 12, wherein the fitness function describes how similar the channel parameters estimated along the h^{th} trajectory by the prediction model, which uses current obstacles parameters, is to gathered channel parameters by a mobile node along its truth trajectory.

14. A location and tracking system as claimed in claim 13, wherein the fitness function takes the form of a set of linear equations for an optimisation method.

15. A location and tracking system as claimed in any preceding claim, wherein the processor is adapted to execute a particle filter algorithm to locate a mobile node in step (c), in which particles are mapped to the physical environment as represented by the physical environment model.

16. A location and tracking system as claimed in claim 15, wherein the processor is adapted to generate visual displays of the physical environment in which particles distribution is illustrated for user visualisation of likelihoods of locations of the mobile node.

17. A location and tracking system as claimed in claims 15 or 16, wherein the processor is adapted to eliminate particles in successive iterations according to constraints imposed by the physical environment model and a motion model, in which remaining particles converge around the actual mobile node location.

18. A location and tracking system as claimed in any of claims 15 to 17, wherein the processor is adapted to automatically recognise rooms in the physical environment model and to maintain a count of particles in each room to provide a per-room probability of presence of the mobile node.

19. A location and tracking system as claimed in any of claims 15 to 18, wherein the processor is adapted to provide a random or uniform initial particle distribution mapped into the physical environment model, to filter the initial particle distribution with a likelihood observation function to generate a second particle distribution and to use said distribution for a next filtering iteration.

20. A location and tracking system as claimed in claim 19, wherein the likelihood observation function is given by combining densities to obtain a likelihood observation function of measurement p(zt|xt), and is given by:

$$p(z_t|x_t) = \left[\prod_{k\in K} p_{hit}(z_t^k|x_t)\right]^{\frac{1}{|K|}} \cdot \prod_{k\in L} p_{miss}(z_t^k|x_t) \cdot \prod_{k\in M} p_{extra}(z_t^k|x_t) \cdot \beta \qquad (3.27)$$

with

$$\beta = \frac{|K|}{|K| + |L| + |M|}$$

where:

K.................... is the set of measurement in $(z_t \cap z^+_t)$,
L.................... is the set of measurement in $(z^+_t - z_t)$,
M.................... is the set of measurement in $(z_t - z^+_t)$,

..............and the expressions of $|K|$, $|L|$, $|M|$ denotes the number of elements in K, L, M respectively,

......................β represents the weighting factor and

.... k denotes AP identification.

21. A location and tracking system as claimed in any preceding claim, wherein the processor is adapted to perform step (d) by adjusting the current parameters of obstacles towards the optimised parameters of obstacles which influence the channel parameters along the h^{th} trajectory, in which the adjustment is given by:

$$Lw^i_t = Lw^i_{t-1} + \Delta Lw^i_t$$

,

where:

Lw^i_t............... represents i^{th} wall segment loss at time (index of iteration) t,
Lw^i_{t-1}............. represents i^{th} wall segment loss at iteration $t-1$, and
ΔLw^i_t............. denotes the delta of a wall segment loss at time t.

22. A location and tracking method performed by a location and tracking system linked with wireless base stations, and having an input interface, and a processor, wherein the method comprises the steps of:

(a) receiving physical environment inputs and providing an initial physical environment model;

(b) generating an initial channel parameters fingerprint for the environment according to the initial environment model;

(c) monitoring channel parameters of a mobile node in the environment and using said parameters and the fingerprint to locate the mobile node;

(d) modifying the environment model according to said node locating step (c);

(e) updating the fingerprint according to the modification of the environment model in step (d); and

(f) repeating steps (c) to (e) in subsequent iterations.

23. A computer readable medium comprising software code adapted to perform operations of a location and tracking method of claim 22 when executing on a digital processor.

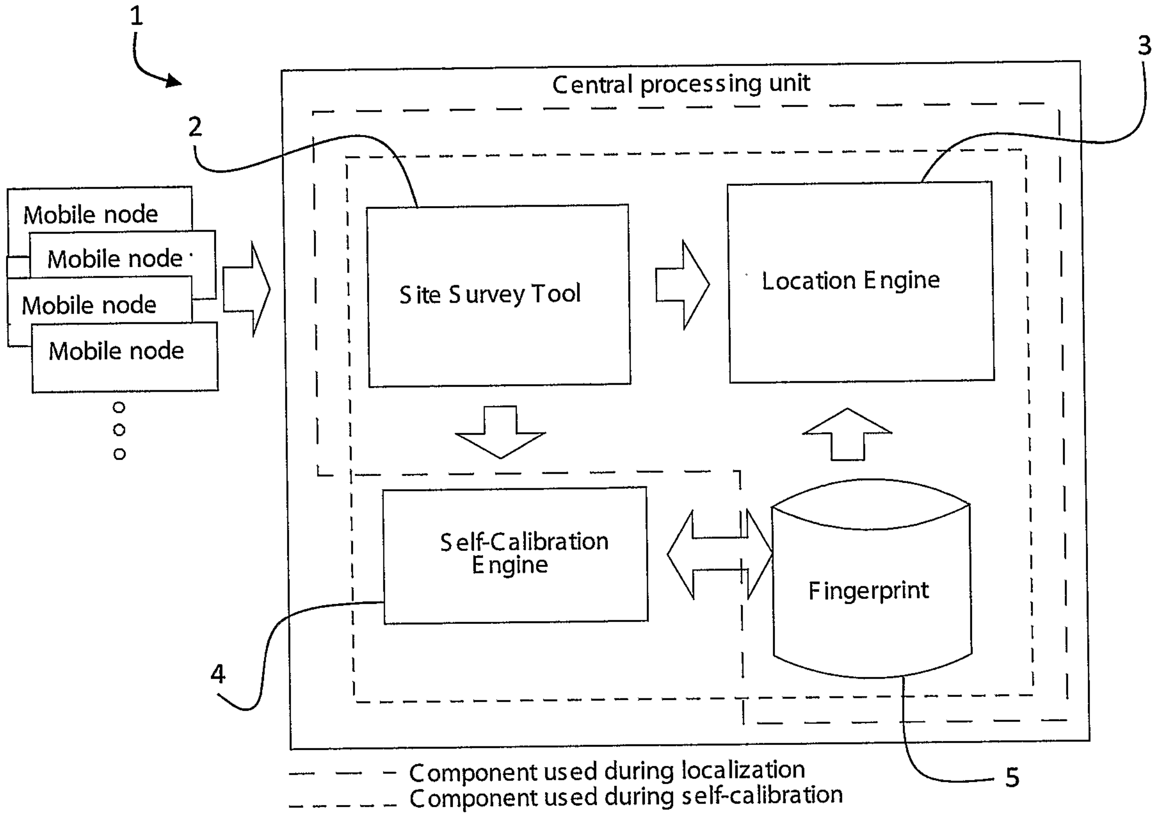

1
2
3
Central processing unit
Mobile node
Mobile node
Mobile node
Mobile node
Site Survey Tool
Location Engine
Self-Calibration Engine
Fingerprint
4
5
Component used during localization
Component used during self-calibration

Fig. 1

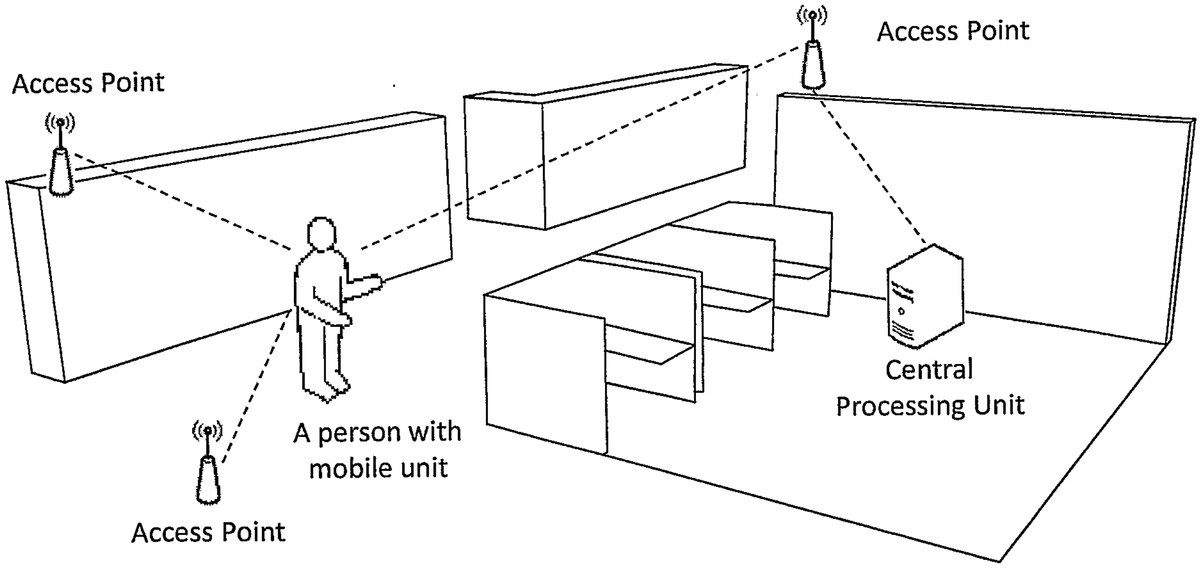

Access Point
Access Point
Access Point
A person with mobile unit
Central Processing Unit

Fig. 2

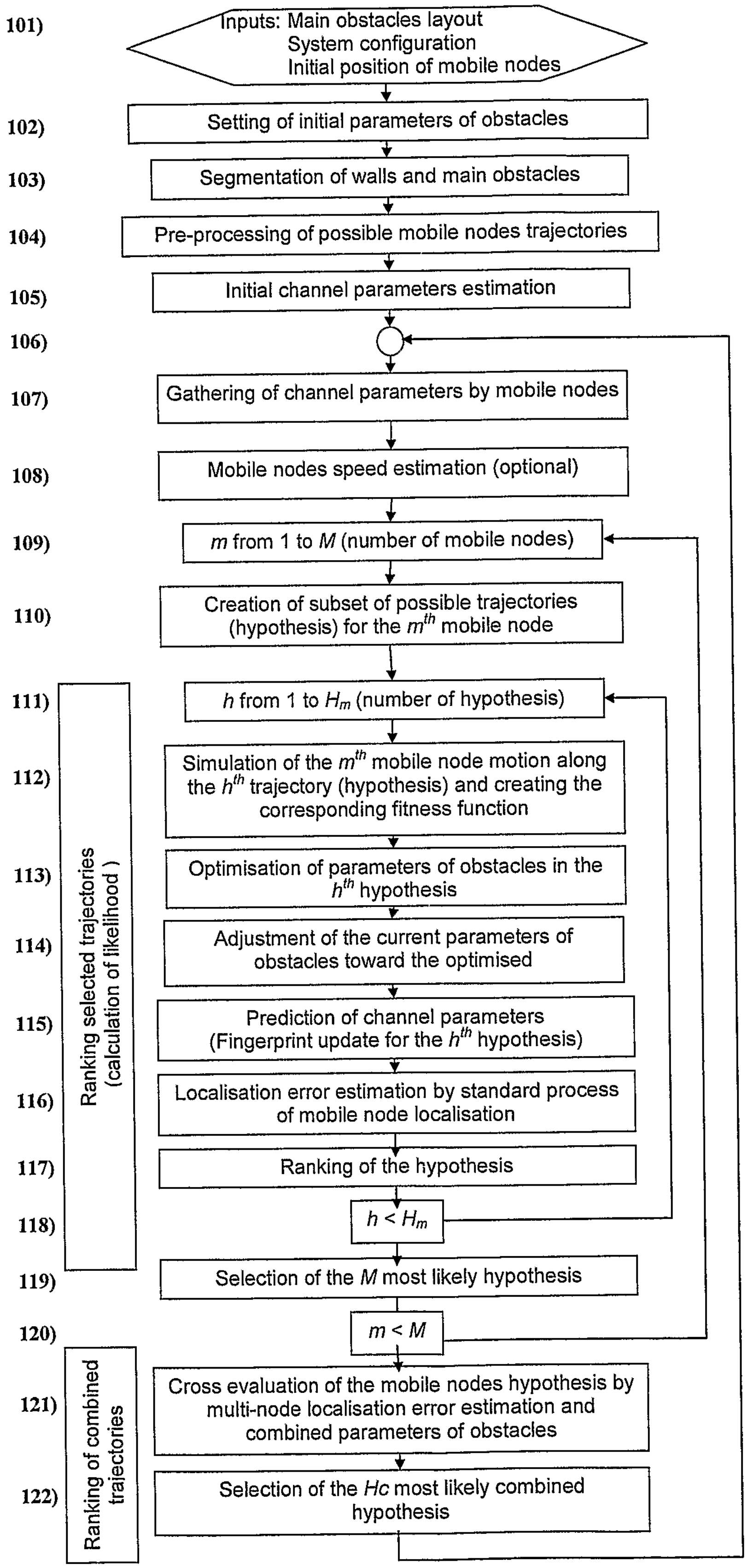

101)
Inputs: Main obstacles layout
System configuration
Initial position of mobile nodes
102)
Setting of initial parameters of obstacles
103)
Segmentation of walls and main obstacles
104)
Pre-processing of possible mobile nodes trajectories
105)
Initial channel parameters estimation
106)
107)
Gathering of channel parameters by mobile nodes
108)
Mobile nodes speed estimation (optional)
109)
m from 1 to M (number of mobile nodes)
110)
Creation of subset of possible trajectories (hypothesis) for the mth mobile node
111)
h from 1 to Hm (number of hypothesis)
112)
Simulation of the mth mobile node motion along the hth trajectory (hypothesis) and creating the corresponding fitness function
113)
Optimisation of parameters of obstacles in the hth hypothesis
114)
Adjustment of the current parameters of obstacles toward the optimised
115)
Prediction of channel parameters (Fingerprint update for the hth hypothesis)
116)
Localisation error estimation by standard process of mobile node localisation
117)
Ranking of the hypothesis
118)
h < Hm
Ranking selected trajectories (calculation of likelihood)
119)
Selection of the M most likely hypothesis
120)
m < M
121)
Cross evaluation of the mobile nodes hypothesis by multi-node localisation error estimation and combined parameters of obstacles
122)
Selection of the Hc most likely combined hypothesis
Ranking of combined trajectories

Fig. 3

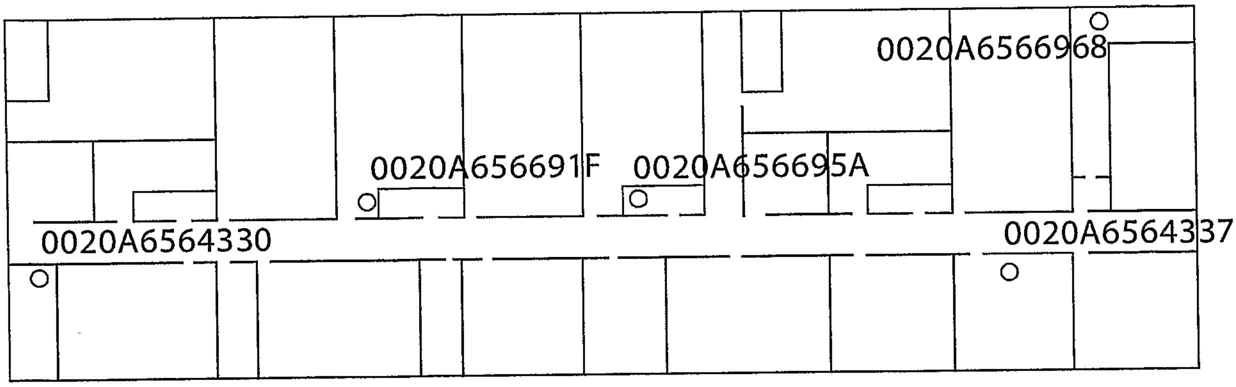
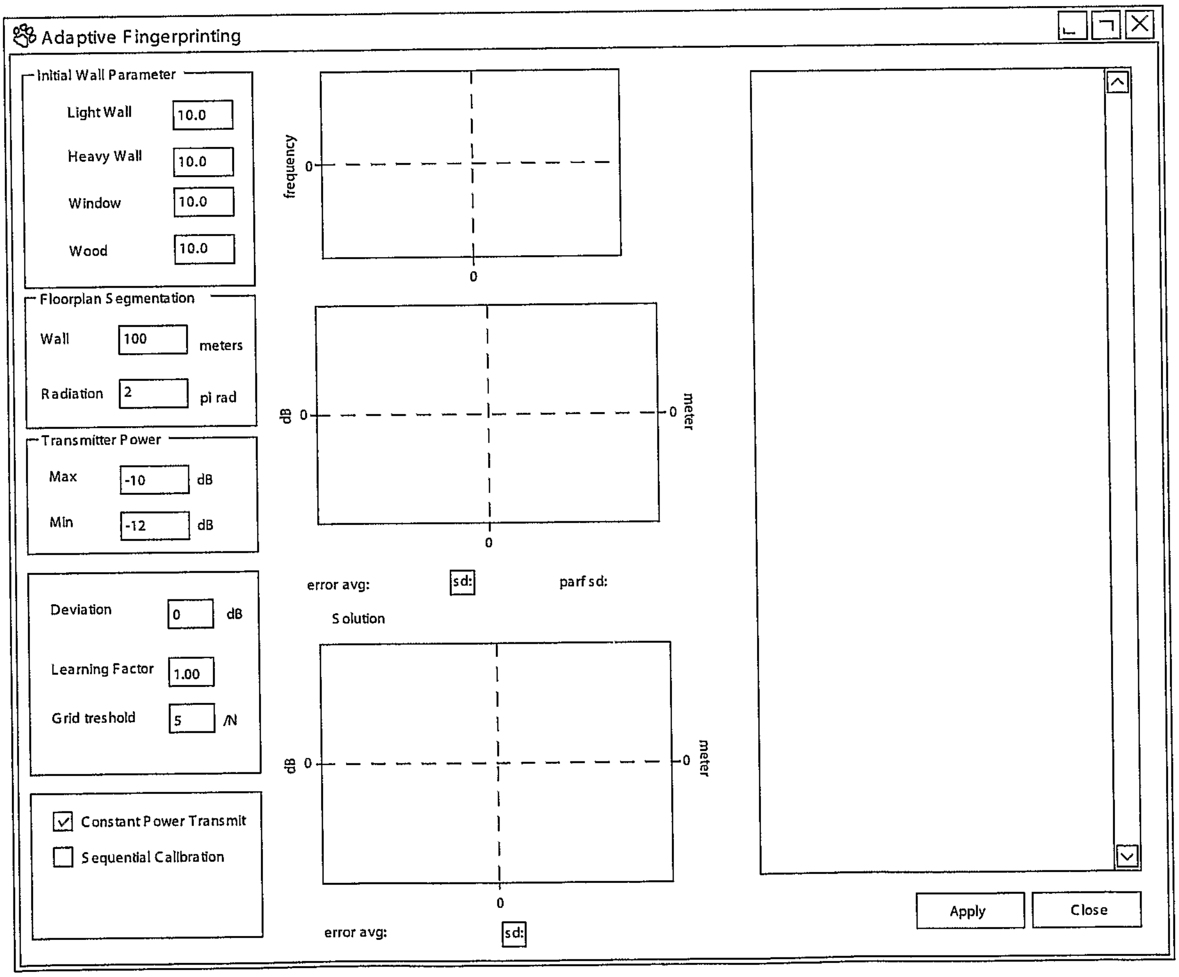


Fig. 4


Adaptive Fingerprinting

Initial Wall Parameter
Light Wall 10.0
Heavy Wall 10.0
Window 10.0
Wood 10.0

Floorplan Segmentation
Wall 100 meters
Radiation 2 pi rad

Transmitter Power
Max -10 dB
Min -12 dB

Deviation 0 dB
Learning Factor 1.00
Grid treshold 5 /N

☑ Constant Power Transmit
□ Sequential Calibration

error avg: sd: parf sd:

Solution

error avg: sd:

Fig. 5

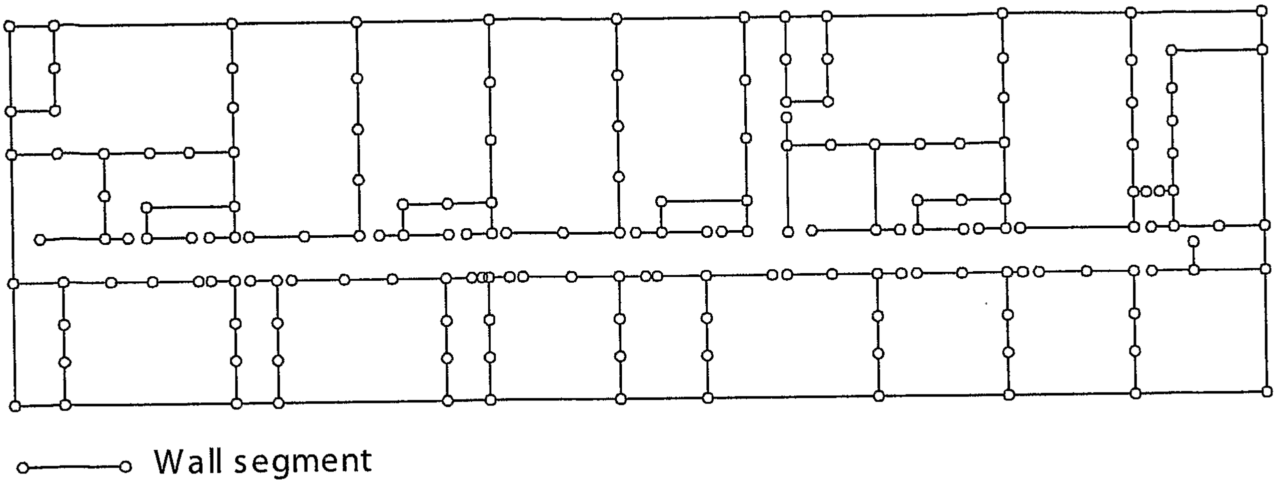


Fig. 6

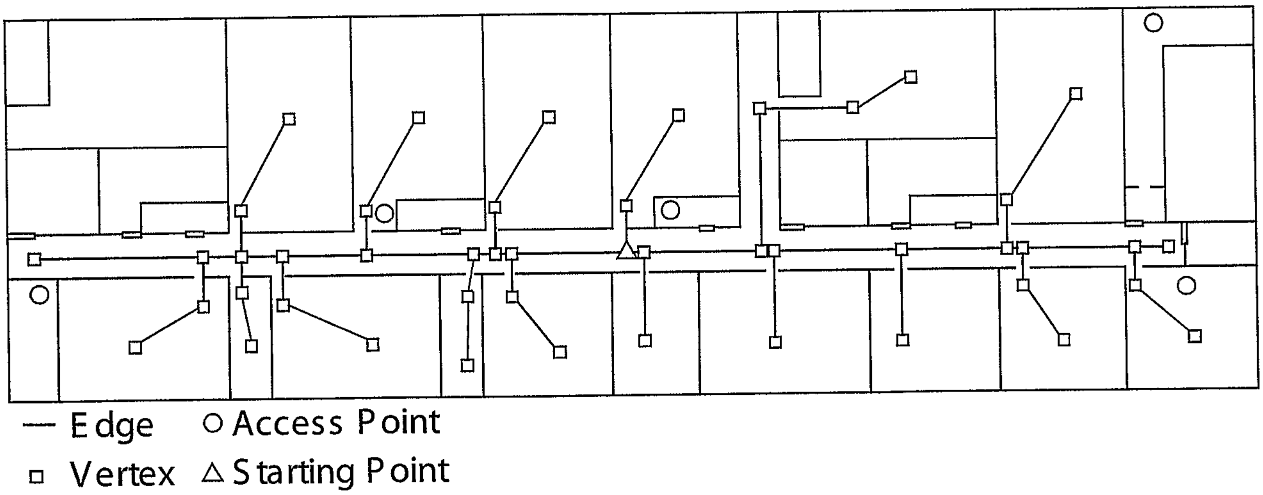


Fig. 7

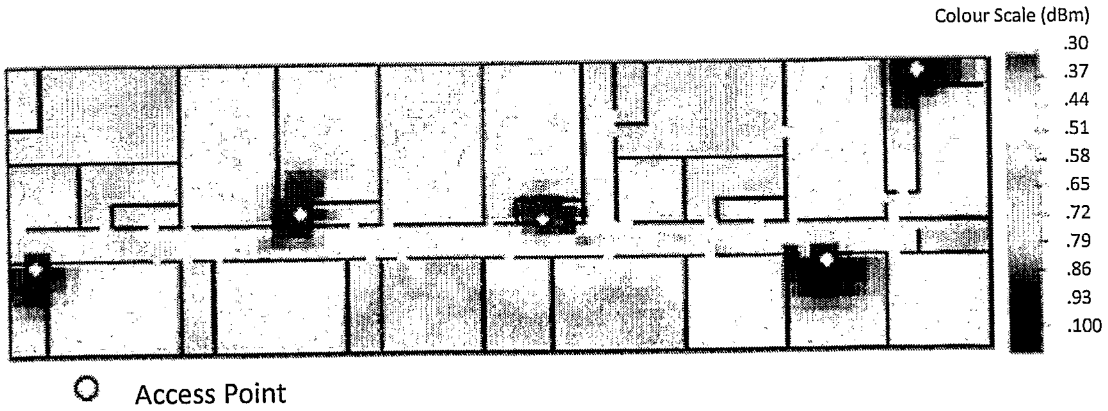


Fig. 8

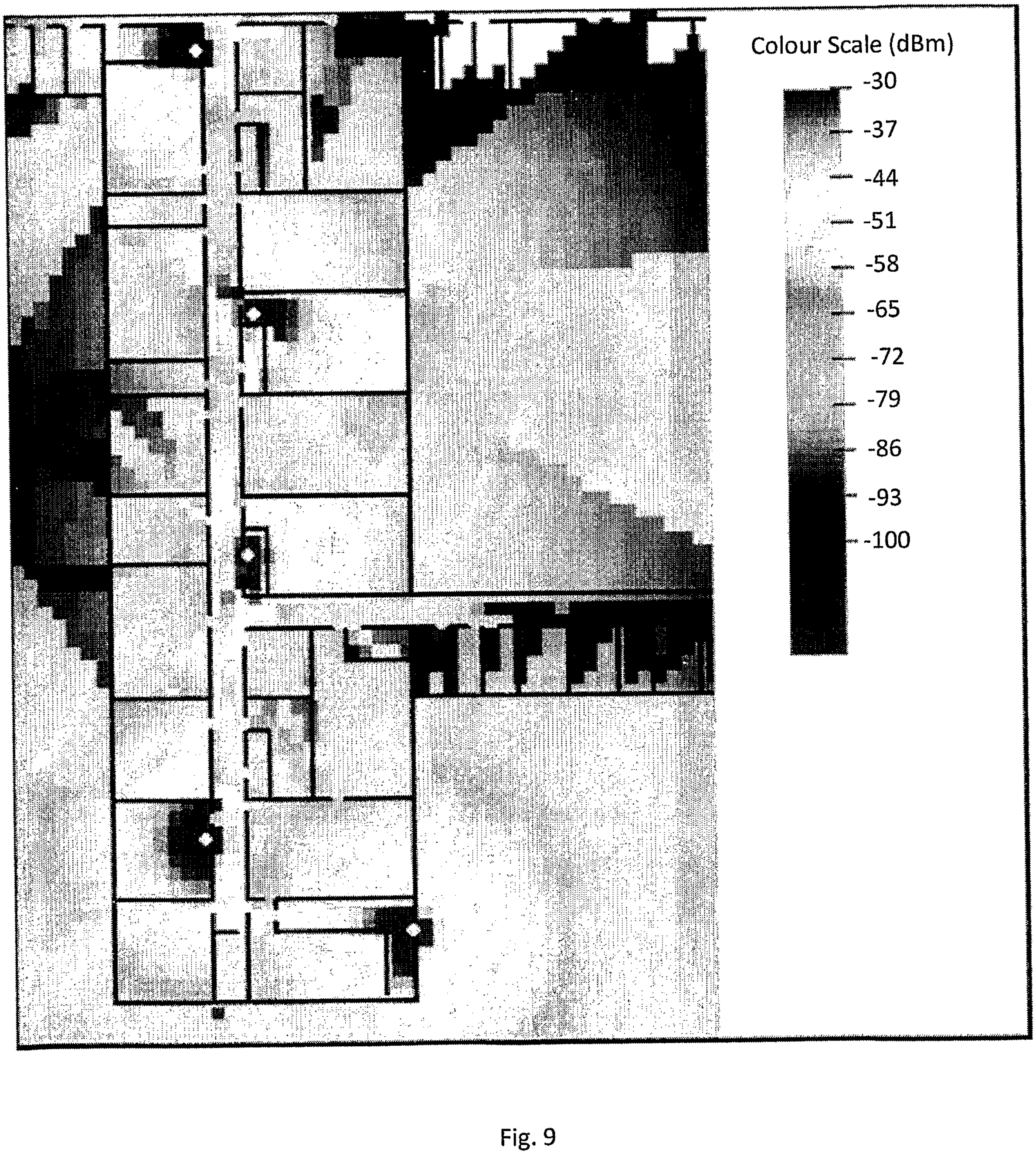
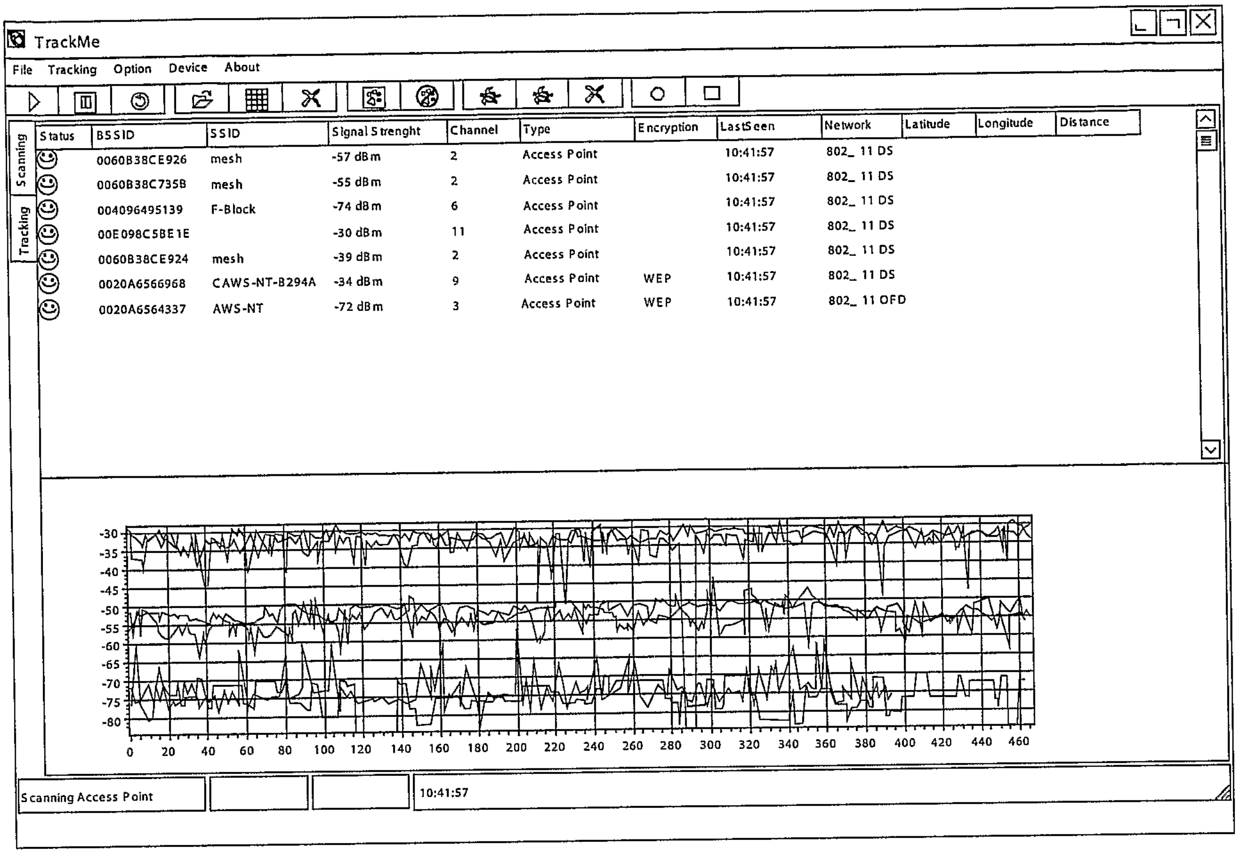


Fig. 9



TrackMe

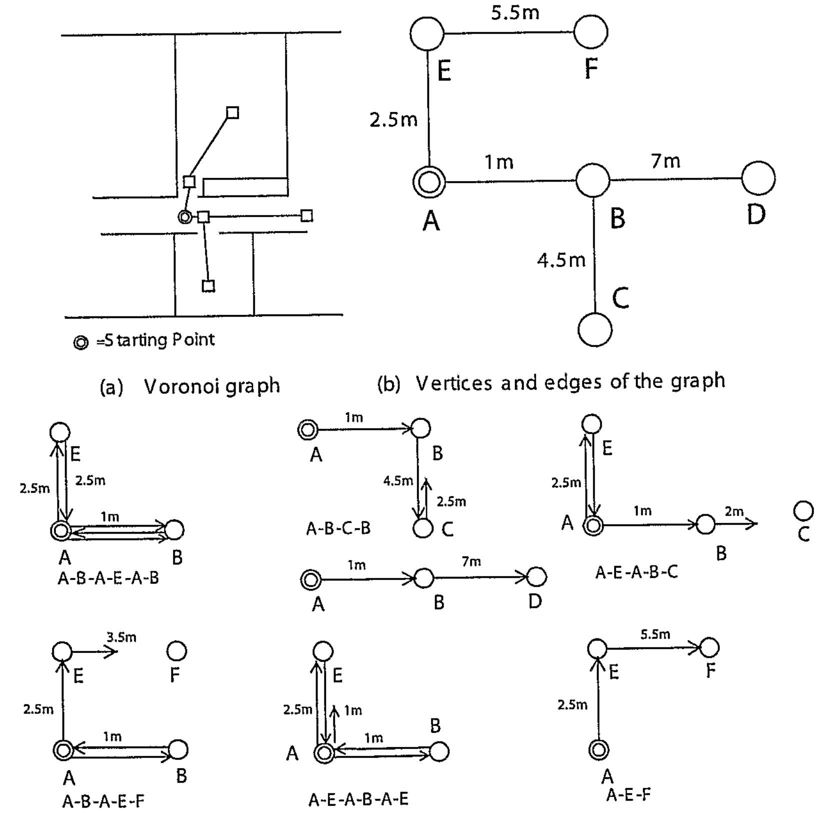
File Tracking Option Device About

Status	BSSID	SSID	Signal Strenght	Channel	Type	Encryption	LastSeen	Network	Latitude	Longitude	Distance
☺	0060B38CE926	mesh	-57 dBm	2	Access Point		10:41:57	802_ 11 DS			
☺	0060B38C735B	mesh	-55 dBm	2	Access Point		10:41:57	802_ 11 DS			
☺	004096495139	F-Block	-74 dBm	6	Access Point		10:41:57	802_ 11 DS			
☺	00E098C5BE1E		-30 dBm	11	Access Point		10:41:57	802_ 11 DS			
☺	0060B38CE924	mesh	-39 dBm	2	Access Point		10:41:57	802_ 11 DS			
☺	0020A6566968	CAWS-NT-B294A	-34 dBm	9	Access Point	WEP	10:41:57	802_ 11 DS			
☺	0020A6564337	AWS-NT	-72 dBm	3	Access Point	WEP	10:41:57	802_ 11 OFD			

Scanning Access Point

10:41:57

Fig. 10



◎ =Starting Point

(a) Voronoi graph

(b) Vertices and edges of the graph

Fig. 11

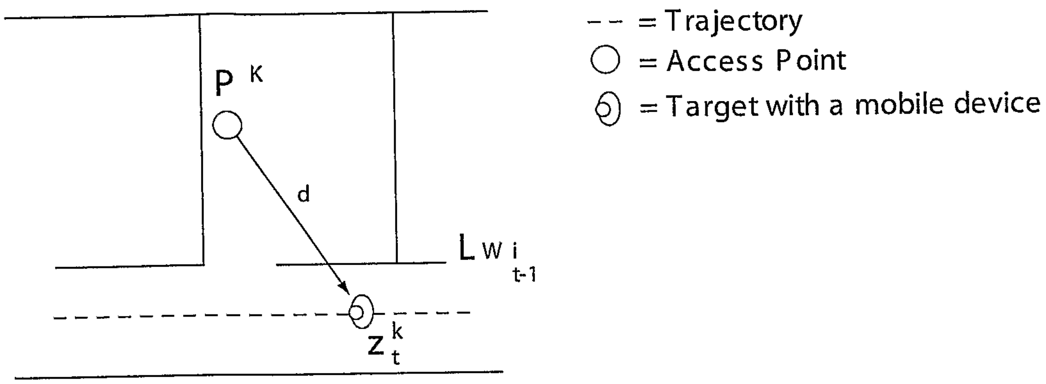

P K
d
L w i t-1
z k t
-- = Trajectory
○ = Access Point
= Target with a mobile device

Fig. 12

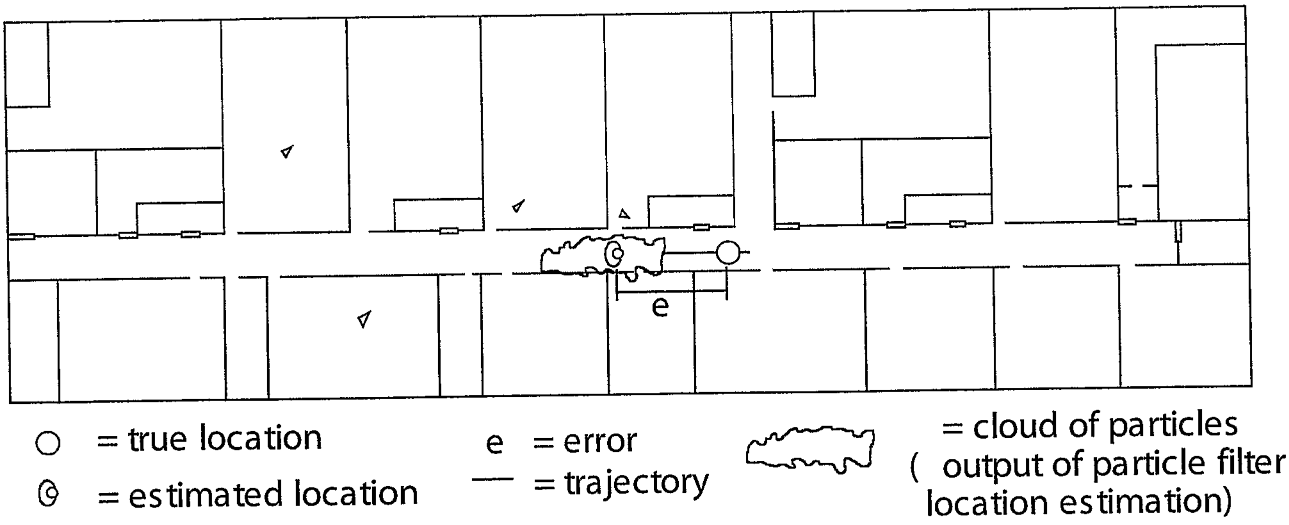

e
○ = true location
= estimated location
e = error
— = trajectory
= cloud of particles (output of particle filter location estimation)

Fig. 13

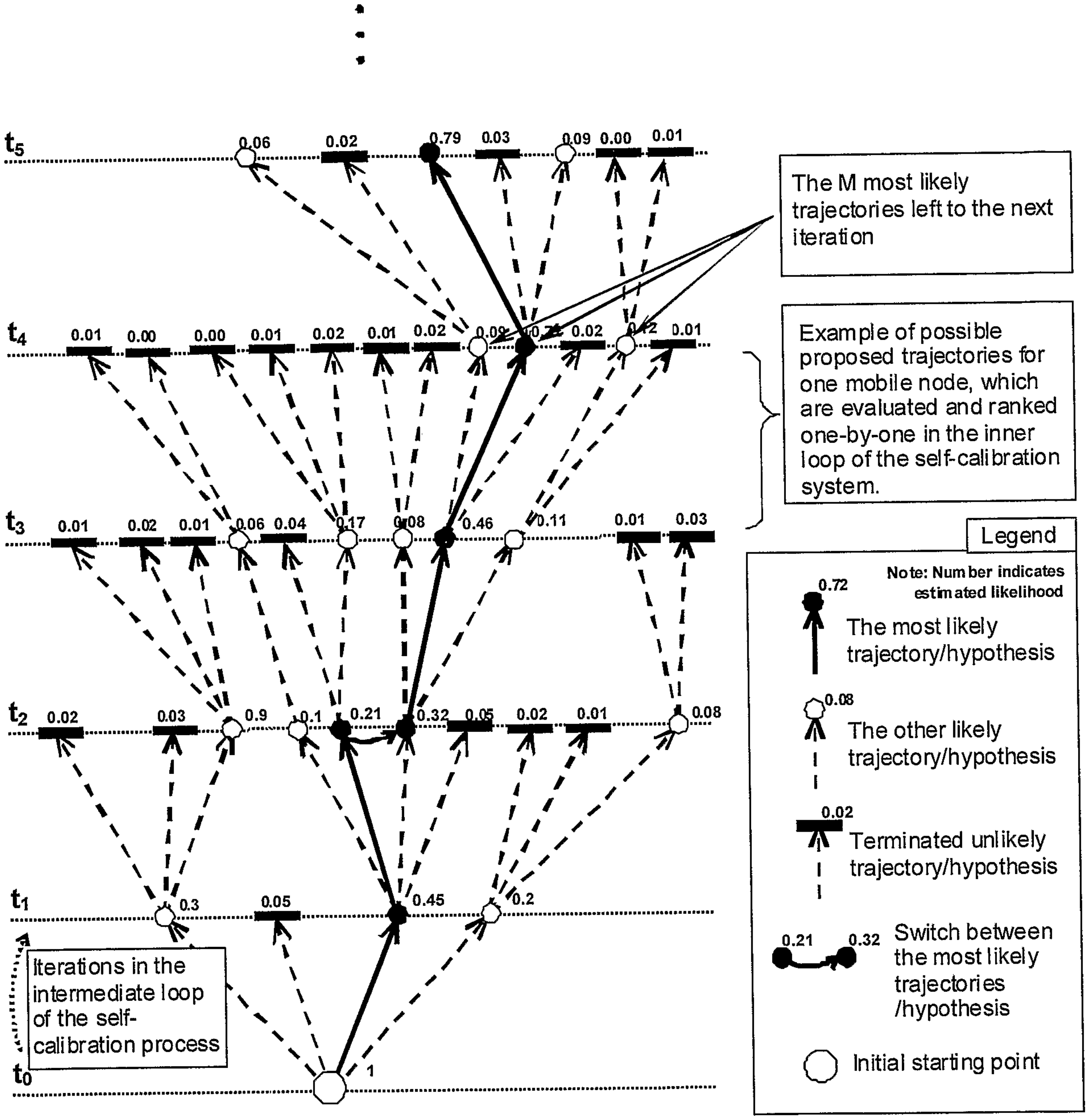

t5
t4
t3
t2
t1
t0
0.06
0.02
0.79
0.03
0.09
0.00
0.01
0.01
0.00
0.00
0.01
0.02
0.01
0.02
0.09
0.72
0.02
0.12
0.01
0.01
0.02
0.01
0.06
0.04
0.17
0.08
0.46
0.11
0.01
0.03
0.02
0.03
0.9
0.1
0.21
0.32
0.05
0.02
0.01
0.08
0.3
0.05
0.45
0.2
1
The M most likely trajectories left to the next iteration
Example of possible proposed trajectories for one mobile node, which are evaluated and ranked one-by-one in the inner loop of the self-calibration system.
Iterations in the intermediate loop of the self-calibration process
Legend
Note: Number indicates estimated likelihood
0.72
The most likely trajectory/hypothesis
0.08
The other likely trajectory/hypothesis
0.02
Terminated unlikely trajectory/hypothesis
0.21
0.32
Switch between the most likely trajectories /hypothesis
Initial starting point

Fig. 14

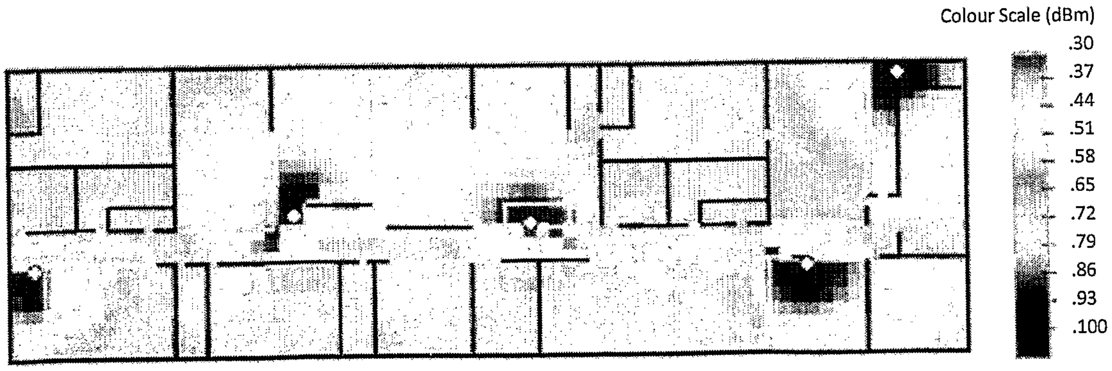
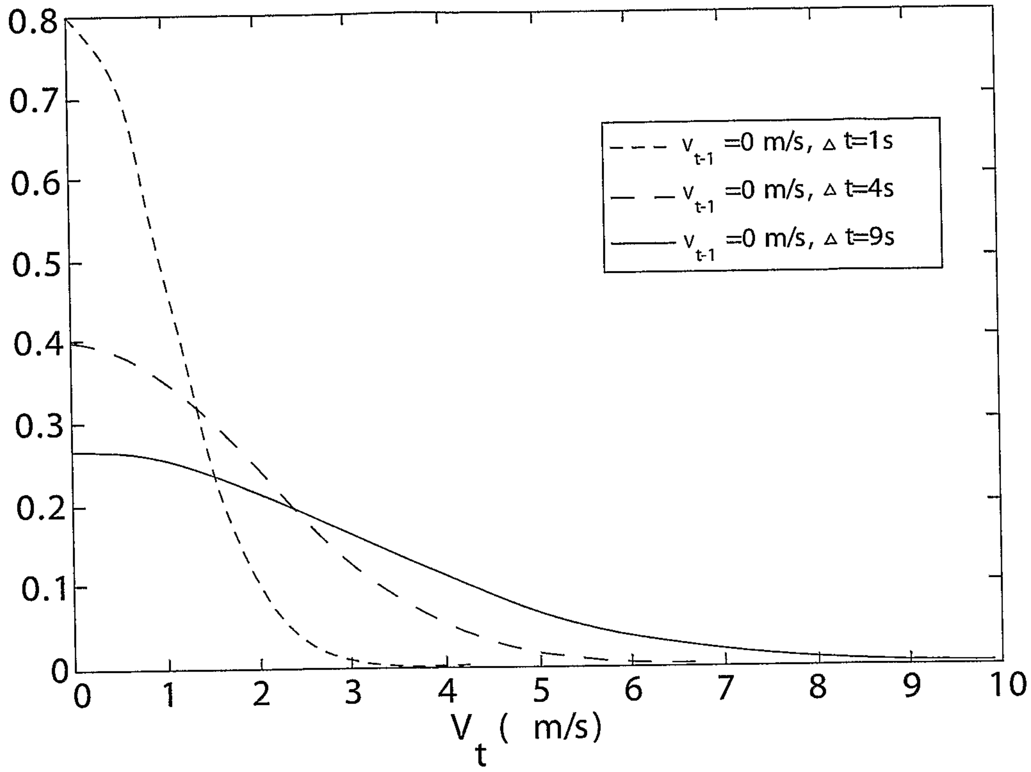
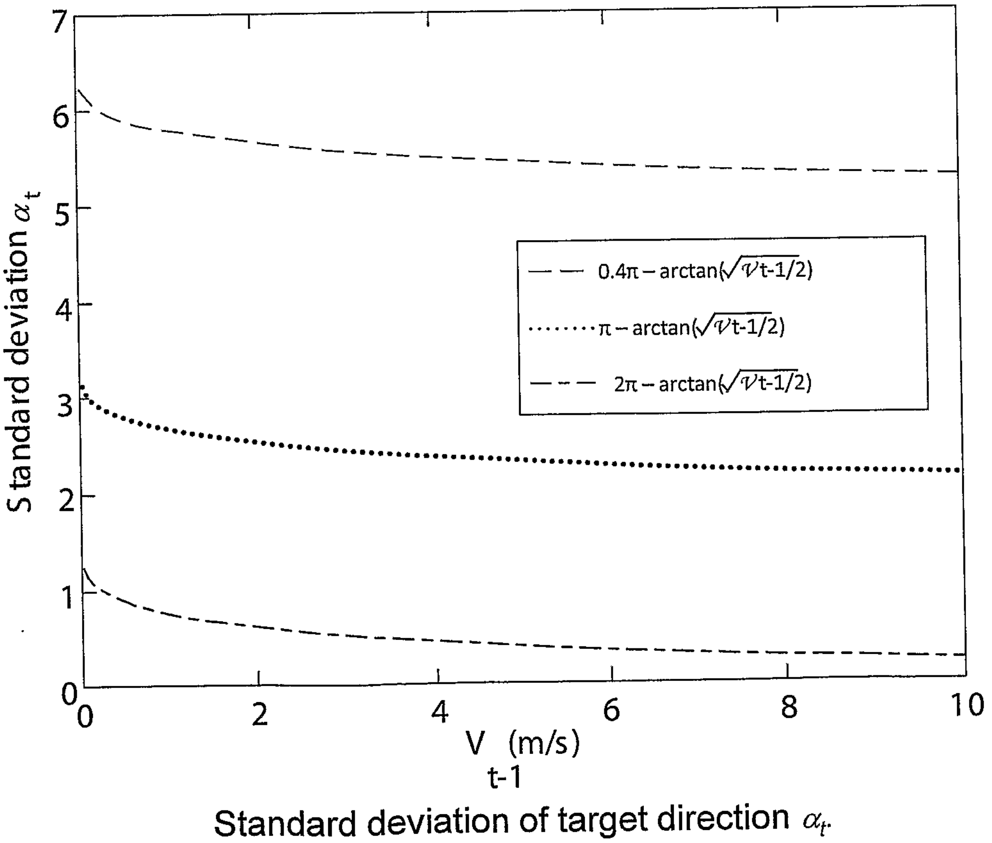

Colour Scale (dBm)
.30
.37
.44
.51
.58
.65
.72
.79
.86
.93
.100

Fig. 15



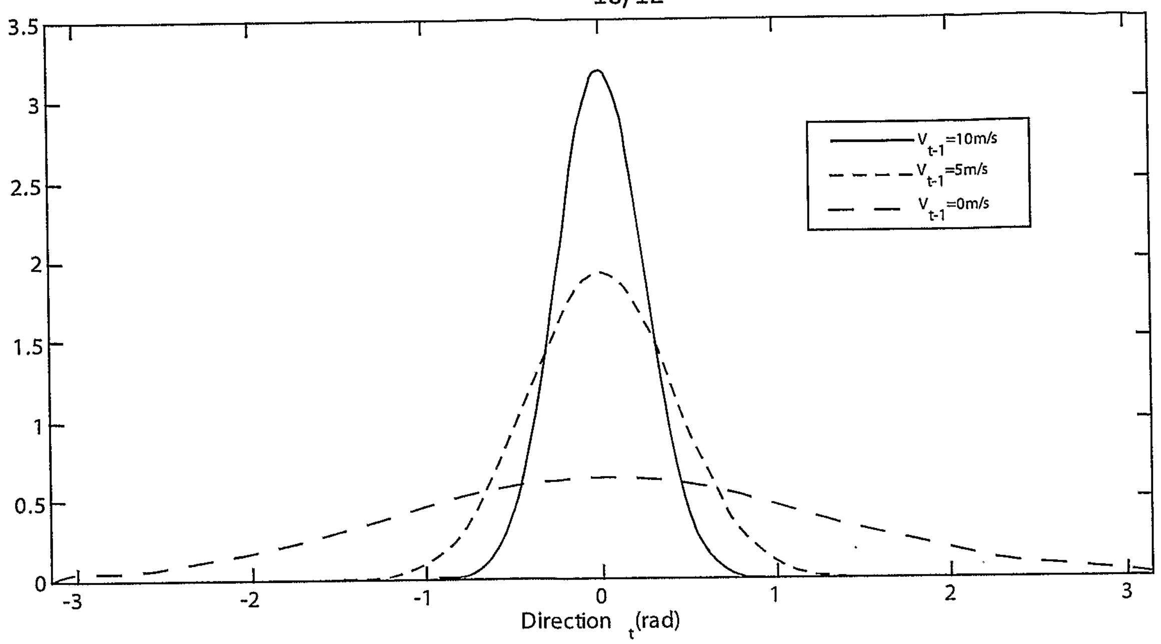
PDF of velocity with v_{t-1} = 0 m/s.

Fig. 16



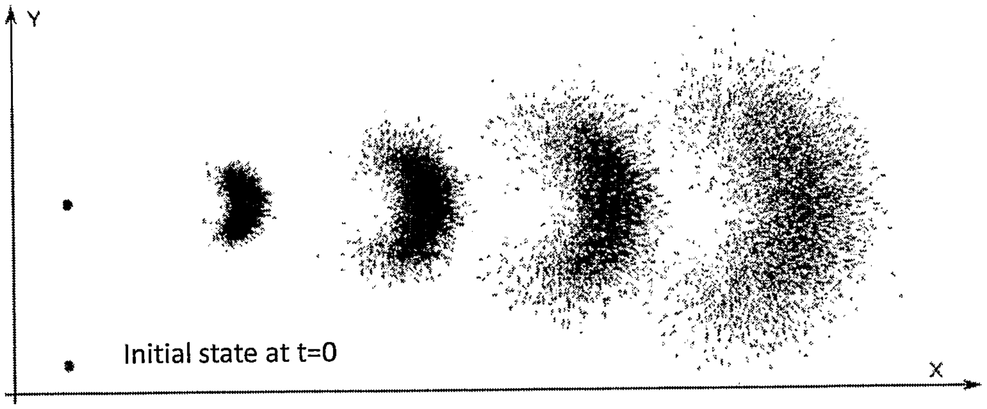
Standard deviation of target direction α_t.

Fig. 17



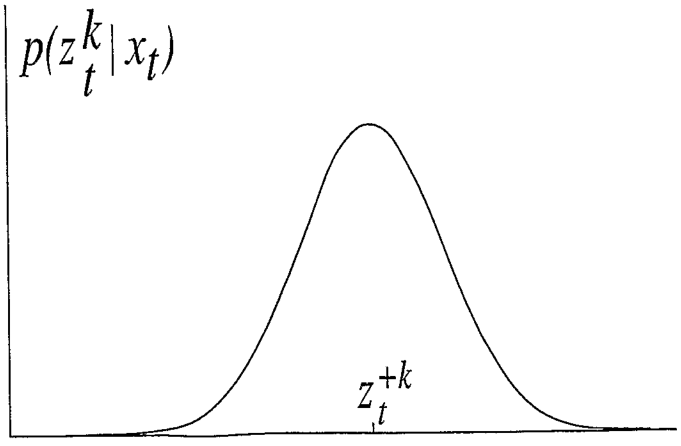
PDF of the direction with $\alpha_{t-1} = 0$.

Fig. 18



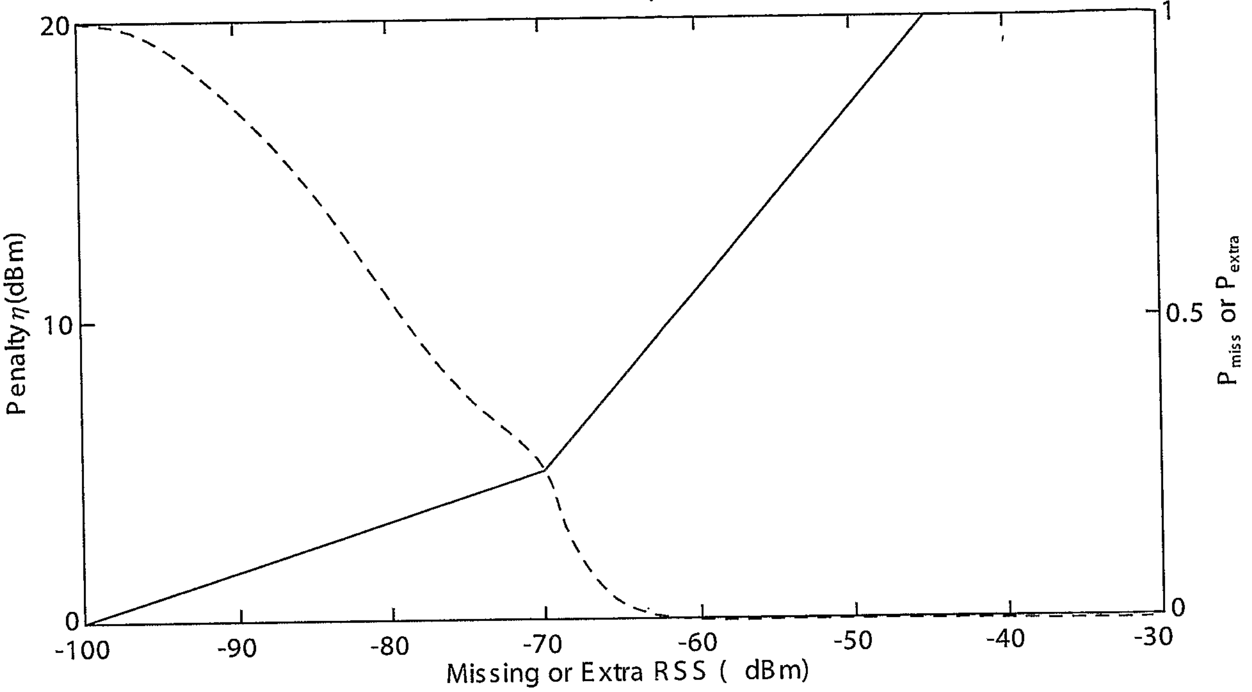
Evolution of particles with motion model.

Fig. 19



Gaussian distribution p_{hit}.

Fig. 20



Penalty η for missing or extra RSS value.

Fig. 21

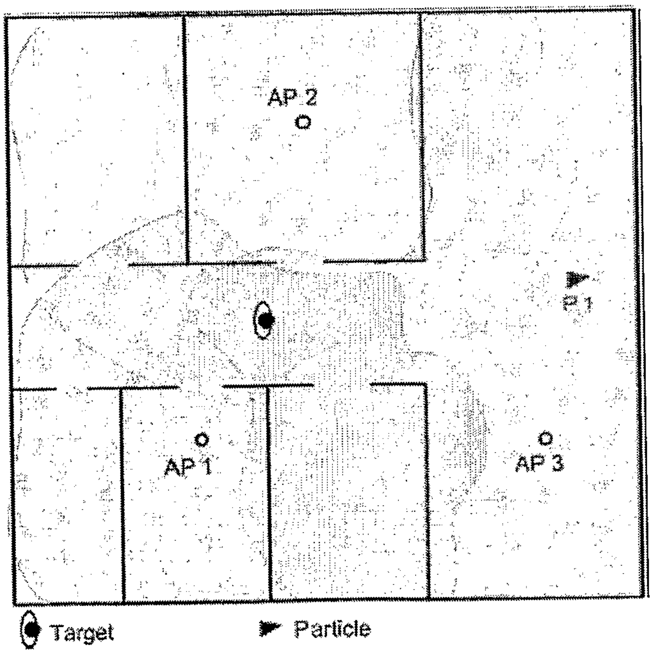
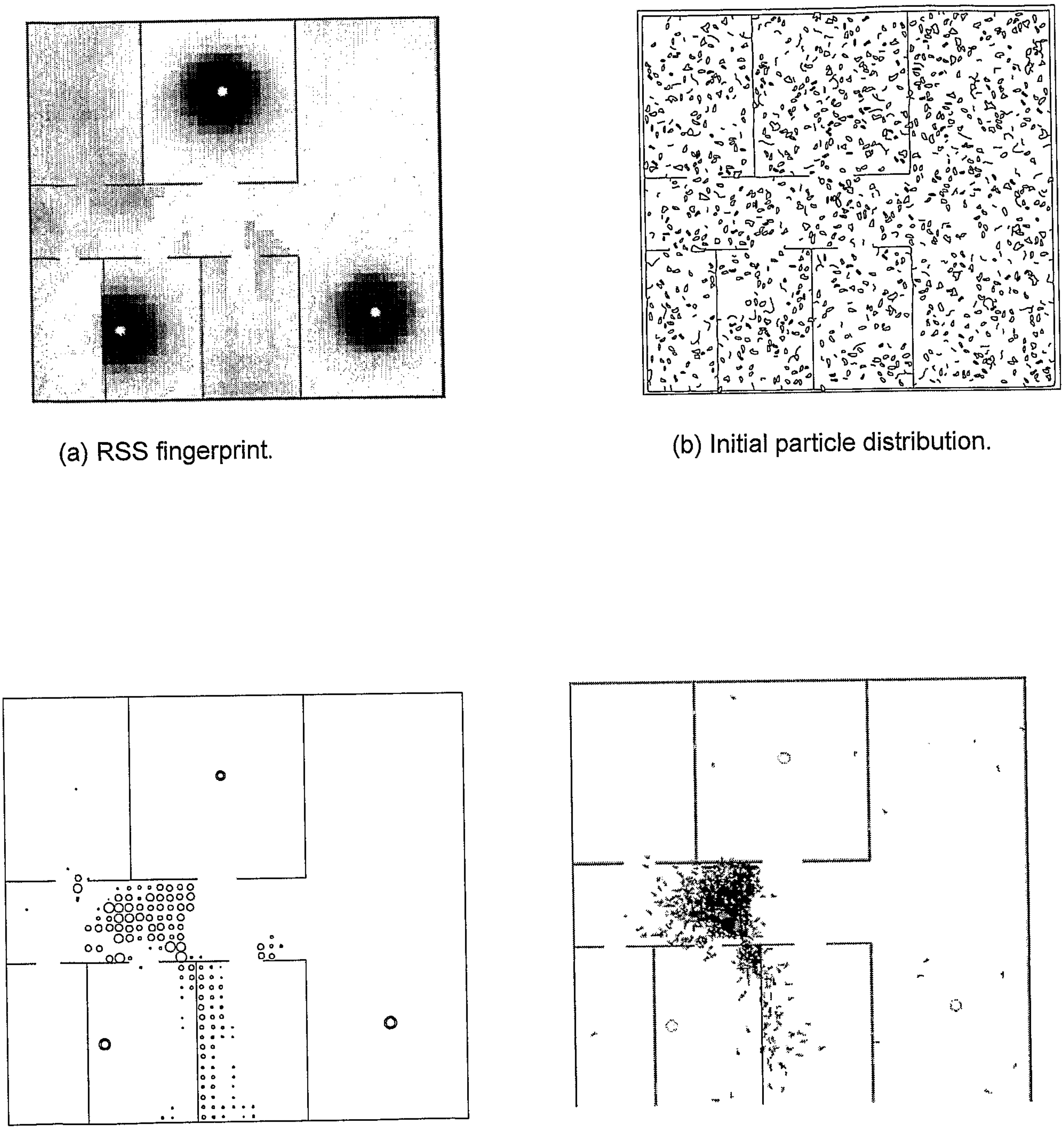


Illustration for the calculation of the measurement model.

Fig. 22



(a) RSS fingerprint.

(b) Initial particle distribution.

(c) Measurement probability.

(d) Updated particle distribution

Update of the particles distribution with the likelihood observation function.

Fig. 23