Fusing Camera and Wi-Fi Sensors for Opportunistic Localization

Sam Van den Berghe, Maarten Weyn
Artesis University College
e-lab
Antwerp, Belgium
Email: sam.vandenberghe@artesis.be
maarten.weyn@artesis.be

Vincent Spruyt
Ghent University
TELIN-IPI-IBBT
Antwerp, Belgium
Email: vspruyt@telin.ugent.be

Alessandro Ledda
Artesis University College
e-lab
Antwerp, Belgium
Email: alessandro.ledda@artesis.be

***Abstract*—There has been a lot of research done towards both camera and Wi-Fi tracking respectively, both these techniques have their benefits and drawbacks. By combining these technologies, it is possible to eliminate their respective weaknesses, to increase the possibilities of the system as a whole. This is accomplished by fusing the data from Wi-Fi and camera before inserting it in a particle filter. This will result in a more accurate and robust localization system. The measurement model for Wi-Fi data uses a difference feature vector for comparing data to the fingerprint. The images taken from the camera are analysed, and filtered to detect human shapes. In this paper it is proven that an increased accuracy can be achieved by fusing the sensor data of both Wi-Fi and camera.**

Keywords-Tracking; Camera; Background subtraction; Wi-Fi; fingerprint.

I. INTRODUCTION

The need for localization is increasing and so is the range of related possibilities. The increasing availability of mobile applications and social networking has increased the request for context aware applications and services, as well as the possible technologies and solutions. There are multiple ways to track people in a building environment. Some are very accurate like ultra-wide band [1] (UWB), while others require no additional infrastructure [2, p. 24]. But there is not one ideal technology covering all needs. There is always a drawback when using a certain technology [3, p. 72]. By combining these technologies, we can try to remove the negative aspects of each individual method and augment its strengths. This paper proposes an algorithm that combines Wi-Fi localization and static camera tracking. The algorithms differ to other solutions, which are presented in Section II, by fusing the sensor data in the measurement model before calculating an estimated position based the individual technologies.

The main goal is by combining Wi-Fi fingerprint based localization and camera tracking, to increase the accuracy and reliability of the overall system. A static camera is more accurate than Wi-Fi localization, but has blind spots, suffers from occlusion and it is difficult to perform identification. Wi-Fi localization is generally accurate up to room level [3], but requires users to carry a Wi-Fi capable device, this also means that identification is inherent in this form of localization. That means that Wi-Fi alone cannot locate anybody who does not want to be tracked, i.e., does not enable his or her Wi-Fi device.

The purpose of fusing Wi-Fi and video data is to have a smaller localization error in the rooms where there is a camera, in contrast to only Wi-Fi, but still offer room level localization where there are no cameras. This paper will rather focus on preparing the captured images and fusing that data with the Wi-Fi data, than on the localization algorithm and Wi-Fi data. The localization algorithm using Wi-Fi is the same as described by Weyn [2] and will be further discussed in Section III-C.

The first aspect of the vision localization is defined as isolating human figures in the image, modelling those areas in the image as a Gaussian mixture model [4] on a floor plan. The fusion of camera and Wi-Fi data will encompass the way the probabilities of both methods are combined to get the most accurate yet still robust tracking.

First the methods that are used will be described, followed by the results attained by these methods. Finally the proposed algorithm and possible future work is discussed.

II. STATE OF THE ART

The research that has been done on the subject of indoor localization using wireless signals is vast as shown by Torres-Solis *et al.* [3]. Methods, such as lateration, angulation and proximity, can be used for localization, but they require the location of the terminals to be known.

Various wireless technologies have been used such as Radio-frequency based localization [5], using Wi-Fi such as RADAR [6], OSL [2]. Other technologies include UWB [1], ultrasound [7] or visual tracking [8]

Oskiper *et al.* [8] combine camera measurement with RF ranging measurements using a Kalman filter. Gee *et al.* [9] combine camera, GPS and UWB. They both use accurate UWB ranging measurements which implies the installation of anchor nodes. Our proposed methods uses the already available Wi-Fi infrastructure to enable opportunistic localization.

 ISBN: 978-1-61208-171-7

Vinyals *et al.* [10] propose a method to combine Wi-Fi and audio measurements. Both measurements are done by the mobile devices which still not solves the security problem since anyone can inactivate their device. The combination of Wi-Fi and fixed cameras enables the use of opportunistic Wi-Fi localization, augmented with cameras placed in the important areas where intruders should be detected.

III. METHODS

In this section the used methods are described, starting with an explanation of particle filters. Afterwards the different measurements and sensor data are explained.

A. Particle Filter

A particle filter [11] is able to cope with the multi-modal nature of the problem, since we can alter the measurement model as desired, depending on the kind of sensor data. An additional problem which can easily be handled using a particle filter is the difference in measurement times. The camera updates multiple times a second while we we only receive every few seconds a Wi-Fi measurement.

The Bayes' rule (Equation 1) explains the reasoning behind a particle filter. To estimate the posterior probability, starting from x being the location and z being the measurement. Since $1/P(z)$, the probability of measurement z, is constant it is replaced with the normalization factor α.

$$P(x_t|z_t) = \alpha P(z_t|x_t)P(x_t) \tag{1}$$

The main components in a particle filter are the motion model, measurement model and resampling [2], [11]. The motion model generally consists of rules that govern how the particles can move, these rules are usually modelled to reflect the real world.

The measurement model describes how the measurements from the world are used to assign a weight to particles. The higher the weight of a particle, the higher the believe of this state. All particle weights sum to one, so that the collection of particles can be called a posterior density function.

The resampling step describes how particles are repositioned between frames. Particles with low weights are removed, while particles with high weights are duplicated. This results in a higher particle density in areas with high probability, since those are the areas that are the most interesting to monitor.

B. Heterogeneous Measurements

Both measurements are fundamentally different: where the Wi-Fi measurement compares the signal strength of the client (a tag, smart phone, netbook, etc.) to a database of signal strengths, camera tracking involves detecting an object as it moves through the environment. This means that Wi-Fi does not have problems with identification, since only the object that is being tracked can transmit the data relevant to its localization and by doing so automatically identifies itself. Identification might be easy for Wi-Fi localization, it cannot track an object that does not give its Wi-Fi signal strength.

Camera tracking has much more difficulties to identify what it is tracking, it is not inherent as with Wi-Fi. However it is possible to detect all other objects in the view plane, so that it is possible to track the people who are not being tracked with Wi-Fi or to increase the accuracy by combining the two measurements.

C. Wi-Fi Localization

The measurement model of [2] is used. It uses pattern matching, here the difference feature vector of the received signal strengths (RSS) from multiple Wi-Fi-access points from the measurement is compared with the fingerprint database using a Gaussian kernel method. Penalties are added if access points are missing from the measurement data or extra access points are found in the measurement data. If an access point is visible at the location of the tag but is not represented in the fingerprint of a certain location, then we assume that the fit between measurement and fingerprint is less accurate and vica versa. This is implemented by adding a penalty to the weight, respective to either the RSS of the extra signal or the expected RSS value.

Because fingerprint matching relies on a database with RSS values from the area wherein the tracking will occur, it is necessary to measure those RSS values at certain intervals in space. This is a drawback, because it requires some manual labour, but is preferred to methods like time-of-flight, because it does not require that the location of access points and difficult environment specific propagation models to be known.

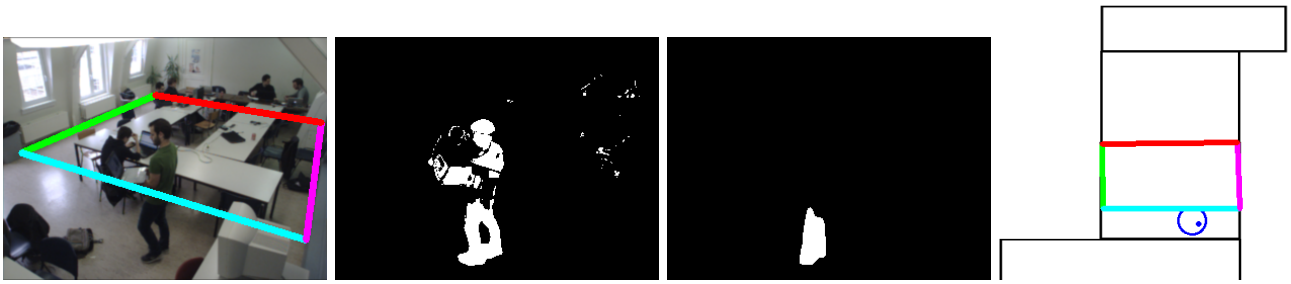
D. Camera Localization System

This section will describe the processing of the video frames before the data is fused together, which is illustrated by Figure 1. First the foreground segmentation is described, followed by how human shapes are extracted and finally mapped to a floor plan.

1) Background Subtraction: Because of the static camera position, a good point to start detecting people is background subtraction. In its most basic form, background subtraction (BGS) takes an image of a room with only background objects, then it uses the absolute difference between the background image and the current video frame, this is called image differencing. After thresholding, this will result in a mask, which segments the foreground objects from the background.

However backgrounds are not static. Changes in lighting and objects being moved, like chairs and tables, can render the background image outdated and useless. To combat this it is necessary to update the background image at a specific learning rate. This results in a trade-off between coping with fast changing environment factors, such as lighting,

 ISBN: 978-1-61208-171-7



(a) (b) (c) (d)

Figure 1. The steps of the visual preprocessing. **(a)** The original image. **(b)** The foreground mask returned by the background subtraction. **(c)** Human filtering applied to the foreground mask.**(d)** The Gaussian kernel of the blob in image (c) mapped to the floor plan.

and preventing temporarily stationary foreground objects to be absorbed in the background. One such method is median background subtraction where the median value of the last n values is used as background model.

An approach that differs from the image differencing in the way that it does not use a single image as background model, is Mixture of Gaussians, which is displayed in Figure 1(b). Here a pixel in the background model is represented by Gaussian kernels at a certain color vector, in this case the RGB color value. Because a pixel can consist of multiple Gaussians, this method can accurately model regions where the background image changes over time between a couple of color vectors, such as a tree branch moving in the wind. a pixel from the current frame is compared to that pixel in the background model, which is a certain amount of Gaussian kernels. If it lies within a certain threshold of a Gaussian it is classified as background. If the pixel that is being compared falls outside all Gaussians it is classified as foreground model and the background model is updated [4].

The resulting image is called a foreground mask, it is basically a binary map of pixels, which are deemed to be of a foreground object. This mask will consist of all objects that are not stationary. This also includes things like chairs that have recently been moved. Since the goal is to track human beings we try to eliminate these false positives. Generally a person will appear as a tall blob in the foreground mask, thus by focusing on these shapes we can reduce the impact of objects like moved furniture. Figure 1(b) shows the result from a mixture of Gaussians BGS.

2) Human filtering: A person in three dimensional space will occupy a cuboid, when projected onto a two dimensional plane, like an image, that person will occupy a rectangle in the image. The image is filtered by a box-filter with the width and height of the rectangle a person would occupy in the image. The difference is that the filter is not centred around its origin point. The origin point is located at the bottom of the structure element, this focuses the most intensity at the bottom of the blob as described by Van Hese [12]. This causes that only blobs, which could be people return a high response, effectively filtering out noise.

An added constraint is that the pixel value at the origin point of the structuring element, has to be higher than a certain threshold. This is done to prevent the filter from returning high values below the detected blob. As a person gets closer to the camera, the region he occupies will get larger as well. This is taken into account by defining two sizes of filter, one at the furthest region in the image and one size for the nearest region, for the rest of the image the size is interpolated between the large en the small size.

The size scaling described in the previous paragraph is preferred above scale space implementation. Scale space estimates the probability of the depth value of a certain object [13], but this consumes a lot of processing power. It scans the entire image multiple times with progressively scaled detection unit, thus creating a three dimensional representation of a two dimensional image. This is superfluous since the orientation of the floor is known, then we can estimate the possible depth of a person based on its location in the image.

At this stage the foreground mask will consist solely of the lowest region of tall blobs, which we assume are the feet of people in the room. This region will be used to map the location in the camera image to a location on a map of the room using only one camera. The transformation from the camera to the floor plan would cast ‘shadows’, bright areas on a map as a result of the projection onto the floor plan.

3) Gaussian modelling: To further prevent this projection effect, and reduce the consumed bandwidth, the filtered foreground mask is described using Gaussian kernels. The kernels that are used are circular 2D Gaussian functions. To model a binary image with Gaussian functions, we make some assumptions and cut corners. For instance, a binary image is not desired when using a particle filter, a more beneficial shape is in fact a Gaussian curve.

With that in mind it is justified to inaccurately model the binary image with Gaussian functions. Secondly, by choosing circular Gaussian functions we can further reduce the ‘shadow’ effect created by projecting the image. By modelling the foreground mask before it is fitted to the floor plan, we can maintain the circular nature of the

 ISBN: 978-1-61208-171-7

(a) (b) (c) (d) (e)

Figure 2. The results of Gaussian modelling. **(a)** A test image with white blobs with increasing size. **(b)** The resulting image from the human filtering. notice that the blob in the center is about the same size as the blob on the left, despite their difference in size in the original image. **(c)** Initial state of Gauss modelling algorithm. **(d)** The third iteration of the algorithm. **(e)** The eighth and in this case final iteration

blobs. The image is modelled by Gaussian curves with coordinates x and y and a σ parameters, only its coordinates are completely transformed while the standard deviation is scaled accordingly, resulting in circular Gaussian functions on the floor plan as seen in Figures 1 (d), which is what is desired.

A method for finding Gaussian distributions in data is Expectation Maximization algorithm. Here a number of Gaussian distributions are mapped to the data. The drawback of this is that the number of separate clusters has to be known, this is not feasible in this set-up. Thus a separate algorithm is devised as shown in Algorithm 1. The proposed algorithm starts from a binary image, where for every white pixel a Gaussian kernel is added to an array of Gaussian kernels. That Gaussian kernel has the same coordinates as the pixel in the image and a default standard deviation. Then every kernel in that list is compared against every other kernel. If two kernels are not c-separated the kernels are combined, meaning their location is averaged and standard deviation is convoluted according to Equation 2. This is done until no new combinations are made. This method is illustrated in Figure 2.

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2} \tag{2}$$

This algorithm gives Gaussian functions located at places with a high probability of having a person there. The formula of a two dimensional circular Gaussian curve is as shown in Equation 3, with $\sigma = \sigma_x = \sigma_y$. The normalizing constant $\frac{1}{\sqrt{2\pi\sigma^2}}$ is there to insure that the integral of the curve is one, it causes the intensity of the peak to decline as the standard deviation gets larger. Large blobs in the image make for large standard deviations in the gauss kernel that represents it, but the larger the blob, the larger the probability of a person being there. Therefore we can disregard the normalizing constant, knowing that the particle filter normalizes itself after measurement.

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}[(\frac{x-\mu_x}{\sigma})^2 + (\frac{y-\mu_y}{\sigma})^2]} \tag{3}$$

Algorithm 1 Mapping Gauss. Curves to blobs in an Image

for all $pixelvalues \geq threshold$ **do**
  $GaussList \leftarrow newgaussKernel$ {pixelcoord, default σ}
end for

unstable = true
while unstable **do**
  for all $gaussKernelsinGaussList$ **do**
    for all $OthergaussKernel\,in\,GaussList$ **do**
      $Distance = \|gaussKernel - OthergaussKernel\|$
      $Total\sigma = gausskernel.\sigma + Othergausskernel.\sigma$
      if $distance \leq Total\sigma$ **then**
        $Combine(gaussKernel, OthergaussKernel)$
      end if
    end for
  end for
  if $noCombinationsoccured$ **then**
    $unstable = false$
  end if
end while

IV. Fusion

Combining the data from Wi-Fi and video is an important step, here it is attempted to increase the amount of valuable information. While other research first computes the location from the each sensor type separately and then fuses the locations, this proposal fuses the sensor data and then uses all available data for estimating the position [2].

The fusion process is shown in Figure 3. Data measured by the sensors are sent to a data aggregator, this component stores the incoming sensor data. The data aggregator selects which measurement models to use, a Wi-Fi or image measurement model or both. The sensor data is then sent to a fusion engine where the particle filter algorithm is controlled. For instance in the event of both measurement models being used, the fusion engine will ensure that the

 ISBN: 978-1-61208-171-7

correct measurement model is used for the corresponding sensor data. The fusion engine will send the renewed location to a GUI (graphical user interface) on the clients device.

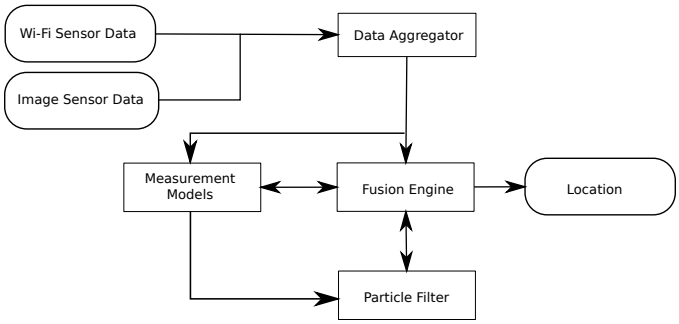


Figure 3. The fusion process flow.

The benefit of fusing these two measurements is that a Wi-Fi measurement only refers to the client while the camera image has data that refers to all persons in its view. A camera provides a sub-meter accurate location but Wi-Fi only has a zone estimation [3]. However the camera has blind spots, is not located in every room, and because of the adaptive background subtraction a stationary person will eventually be absorbed in the background. Therefore it is critical to determine what the state of the sensors are.

Wi-Fi can be used as a stand-alone measurement and will locate a person up to room level, but since this vision system's measurement has no concept of identification it is ill advised to use it as measurement on its own, else there would be no way to determine that the correct person has been located. Additionally, seeing the nature of the transmitted data from a camera server, there is either data or there is not, it is possible to decide which variation of measurement model to use. In the case where only Wi-Fi data is received, obviously only the measurement model for Wi-Fi is used.

When both Wi-Fi and camera data are available, then the two measurements are combined with a naive Bayesian with a confidence measure, as in Equation 4. After this occurrence the same Wi-Fi measurement is repeated when newer camera data is available. Initially, the confidence measure α for the Wi-Fi measurement is one, i.e., very confident since the measurement has just been taken. As the Wi-Fi data becomes older the confidence in that measurement decreases, so that eventually when α is zero, the entire probability, $P(Wi\text{-}Fi|Loc)$ is reduced to 1, and effectively removed from the equation. Similarly, the confidence measure β is determined by the amount and distribution of kernels, where β will be closer to one when there are fewer kernels and these are bunched close together, indicating only one person in the room, and closer to zero when there are a lot of kernels that are spread over a larger area, at the point where the information received from the camera is no longer useful and can in fact be harmful to the localization.

$$P(loc|wifi, cam) = \alpha * P(wifi|loc)^{\alpha} * P(Cam|loc)^{\beta} \quad (4)$$

V. Result

The processing time that it takes for the incoming image to be transformed to the mixture of Gaussians is about 50 milliseconds for an image the size of 640x480 pixels. Taking into account the fact that this only updates at 4 Hz, it leaves the processor with enough time to perform other tasks. The localization engine has an average processing time of 150 milliseconds, this performance number does not change depending on the type of localization that is done, i.e., there is no difference between Wi-Fi, camera or the combination.

The estimated location is compared to the ground truth, and differences in measurement models as to compare the performance of Wi-Fi alone, camera alone and the both combined. The resulting 2 dimensional error is represented as a cumulative distribution function shown in Figure 4. This allows for fast analysis of both the accuracy and precision.

The test itself consisted of one person being tracked in a test environment, shown in Figure 1. The environment consisted of the field of view of a static camera located at the ceiling of the test area, about 3 meters high. The test environment itself is a unmodified lab area with tables and chairs creating occlusion. There are Wi-Fi fingerprints in the test area, but not at every location, because the layout of the tables was different when the fingerprints were taken. The test person walks around at a steady pace, sometimes stopping and changing direction.

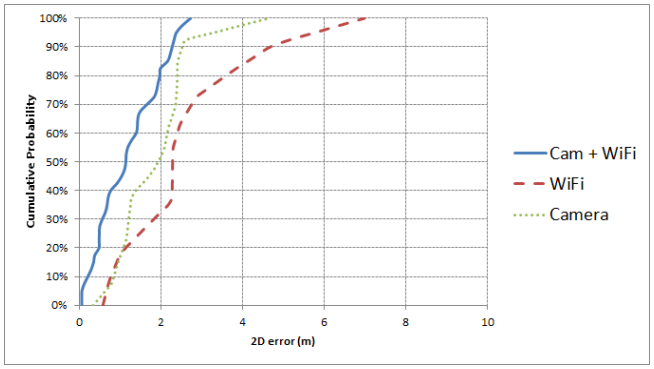
The conditions that were tested included a person with a Wi-Fi client moving around in the test area alone with no interference, this situation is represented by Figure 4(a). Other conditions include a stationary Wi-Fi client while a person walks around an important test since the background subtraction algorithm will not detect someone who has been stationary for a while. Also a cluttered scene where one Wi-Fi client and several others walk in the test area, this can cause problems because the camera sensor data has no identification this can be slightly countered by the confidence measure. The cumulative distribution function of these other situations is shown in Figure 4(b), and displaying a slight increase in accuracy to Wi-Fi.

VI. Conclusion

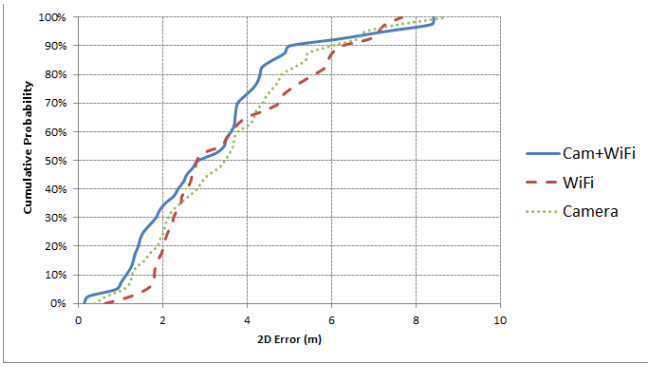
The results indicate that by combining Wi-Fi and camera sensor data, the accuracy can be increased. This is caused by the combination Wi-Fi having only room level accuracy and camera having no concept of identity.

There is also the added benefit of being able to update the clients location faster, than using Wi-Fi alone. This can be vital when trying to guide a person through a building, if the

 ISBN: 978-1-61208-171-7



(a)



(b)

Figure 4. **(a)** The cumulative distribution function of the user walking around the test area without interference.**(b)** all other situations combined

location displayed is several seconds old than it is difficult for that person to orientate him- or herself.

Furthermore because of a fairly accurate measurement from the camera, it is possible to update a Wi-Fi fingerprint if the location provided by the camera is certain enough. Furthermore it is also a possibility to auto-calibrate the camera, meaning that it is possible to place the camera in a specific room by measuring the probabilities of multiple hypotheses of camera locations.

It would also be possible to have a feedback to the camera server on the identity of a kernel, were every kernel has a hypothesis on the identity that it represents [14].

REFERENCES

[1] R. Zetik, J. Sachs, and R. Thoma, "UWB localization - active and passive approach [ultra wideband radar]," in *Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE*, vol. 2, may 2004, pp. 1005 – 1009 Vol.2.

[2] M. Weyn, "Opportunistic Seamless Localization," Ph.D. dissertation, University of Antwerp, Mar. 2011.

[3] J. Torres-Solis, T. H. Falk, and T. Chau, *A review of indoor localization technologies: towards navigational assistance for topographical disorientation*. In-Tech Publishing, 2010, ch. 3, pp. 51–84.

[4] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4. Ieee, 2004, pp. 3099–3104.

[5] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies, "Indoor Wi-Fi positioning: techniques and systems," *Annals of Telecommunications*, vol. 64, pp. 651–664, 2009, 10.1007/s12243-009-0122-1.

[6] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000, pp. 775 –784 vol.2.

[7] A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha, "Tracking Moving Devices with the Cricket Location System," in *2nd International Conference on Mobile Systems, Applications and Services (Mobisys 2004)*, Boston, MA, June 2004.

[8] T. Oskiper, H. Chiu, Z. Zhu, S. Samarasekera, and R. Kumar, "Multi-modal sensor fusion algorithm for ubiquitous infrastructure-free localization in vision-impaired environments," in *IROS*. IEEE, 2010, pp. 1513–1519.

[9] A. Gee, A. Calway, and W. Mayol-Cuevas, "Visual Mapping and Multi-modal Localisation for Anywhere AR Authoring." in *the ACCV Workshop on Application of Computer Vision for Mixed and Augmented Reality*, November 2010.

[10] O. Vinyals, E. Martin, and G. Friedland, "Multimodal indoor localization: An audio-wireless-based approach," in *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*. IEEE, 2010, pp. 120–125.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *Signal Processing, IEEE Transactions on*, vol. 50, pp. 425 – 437, 2002.

[12] P. Van Hese, S. Gruenwedel, V. Jelaca, J. Nino, and W. Philips, "Evaluation of Background/Foreground Segmentation Methods for multi-view Occupancy Maps," in *2nd International Conference on PoCA*, 2011.

[13] R. Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, june 2003, pp. II – 234–40 vol.2.

[14] D. Schulz, D. Fox, and J. Hightower, "People Tracking with Anonymous and ID-Sensors Using Rao-Blackwellised Particle Filters," in *Proc. of the International Joint Conference on Artificial Intelligence*, 2003.

 ISBN: 978-1-61208-171-7