Combining wireless and visual tracking for an indoor environment

Sam Van den Berghe*, Maarten Weyn**, Vincent Spruyt***, Alessandro Ledda****
*Artesis University College, Antwerp, Belgium. Email: sam.vandenberghe@student.artesis.be
**Artesis University College, Antwerp, Belgium. Email: maarten.weyn@artesis.be
***Artesis University College, Antwerp, Belgium. Email: v.spruyt@ieee.org
****Artesis University College, 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 sensor data from Wi-Fi and camera before inserting it in a particle filter. This will result in a more accurate and robust localization system.**

I. Introduction

The need for localization is increasing and so is the range of possibilities. The increasing availability of mobile applications and social networking has made the hunger for localization greater, as well as the possible solutions. There are multiple ways to track people in a building environment. Some are very accurate like ultra-wide band [1], while others require no additional infrastructure [2, p. 24]. But there is not one ideal technology, 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 main goal is by combining Wi-Fi fingerprint 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 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 hers 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 in [2].

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

II. Methods

A. Particle Filter

A particle filter [5] is able to cope with the multi-modal nature of the problem of locating a person in a complex environment. An added benefit of using a particle filter is that we can alter the measurement model as desired, so we can have different measurement models depending on what kind of data is available. To properly scan all the channels for Service Set Identifier (SSID) and signal strength a certain amount of time is needed, while a camera updates multiple times per second. This results in data arriving asynchronous. In case of a high server load it is possible that several frames of image data are ignored.

The main components in a particle filter are the motion model, measurement model and resampling [2], [6]. The motion model generally consists of rules that govern how the particles can move, these rules are usually modeled to reflect the real world. It is possible to move the particles randomly but it is more efficient to move particles as the tracked object would.

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 more 'correct' we estimate that particle to be. The sum of all particle weights need to be 1, 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. Measurement

Both measurements are fundamentally different: where the Wi-Fi measurement compares the signal strength of the client (a tag, smartphone, 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 viewplane, 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

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. The fingerprint data and the measurement taken by a client are compared, using the kernel method. Penalties are added if access points are missing from the measurement data or if extra access points are found in the measurement data. If an access point is visible at the location of the tag or device but is not represented in the fingerprint of a certain location, then we assume that the fit between measurement and fingerprint is less accurate. The same logic applies when there is an access point missing in the measurement that is saved in the fingerprint. 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 labor, but is preferred to methods like time of flight, because it does not require that the location of access points are known.

D. Vision system

This section will describe the processing of the video frames undergo before the data is fused together which 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, and preventing temporarily stationary foreground objects to be absorbed in the background.

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 centered 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 et al. [7]. 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.

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. The reason why the lowest region is the most interesting is because that is the most accurate way of transforming the location in the camera image to a location on a map of the room. The transformation from camera to floor plan would cast ‘shadows’, bright areas on a map as a result of the projection onto the floor plan.

3) Gaussian modeling: 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

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

Fig. 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.

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, a peak with a gentle slope.

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 modeling the foreground mask before it is fitted to the floor plan, we can maintain the circular nature of the blobs. The image is modeled 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 Figure 1 (d), which is what is desired.

A method for finding Gaussian distributions in data is Expectation Maximization algorithm [8]. 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 setup. 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, with a standard deviation, is added to an array of Gaussian kernels. Then every kernel in that list is compared against every other kernel. If two kernels are not c-separated [9] the kernels are combined, meaning their location is averaged and standard deviation is convoluted according to equation 1. This is done until no new combinations are made. This method is illustrated in Figure 2.

$$\text{new } \sigma = \sqrt{\sigma_1^2 + \sigma_2^2} \tag{1}$$

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

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 2, 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.

Algorithm 1 Mapping Gauss curves to blobs in an image

```
for all pixelvalues ≥ threshold do
  GaussList ← newgaussKernel {pixelcoord, defaul σ}
end for
unstable = true
while unstable do
  for all gaussKernelsinGaussList do
    for all OthergaussKernel in GaussList do
      Distance = ||gaussKernel − OthergaussKernel||
      Totalσ = gausskernel.σ + Othergausskernel.σ
      if distance ≤ Totalσ then
        Combine(gaussKernel, OthergaussKernel)
        not stable = true
      end if
    end for
  end for
  if noCombinationsOccured then
    unstable = false
  end if
end while
```

III. Fusion

Combining the data from Wi-Fi and video is an important step, here it is attempted to increase the amount of valuable information. The benefit of fusing these two measurements is that Wi-Fi is data that only refers to the client while the camera image has data that refers to all persons in its view. A camera provides a submeter accurate location but Wi-Fi only has a zone [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 as only sensor can be used as a 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. There would be no way

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

Fig. 2. The results of Gaussian modeling. **(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 modeling algorithm. **(d)** The third iteration of the algorithm. **??** Eight and in this case final iteration

to determine that the correct person has been located. Because the nature of the transmitted data from a camera server, there is either current sensor data or there is not, it is possible to decide in real-time 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 3. 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, and closer to zero when there are a lot of kernels that are spread over a larger area.

$$P(loc|Wi\text{-}Fi, cam) = \alpha * P(Wi\text{-}Fi|loc)^{\alpha} * P(Cam|loc)^{\beta} \tag{3}$$

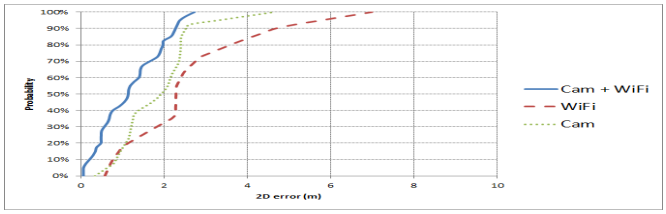
IV. Result

The resulting data are 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 (CDF) shown in Figure 3. This allows for fast analysis of both the accuracy and precision.

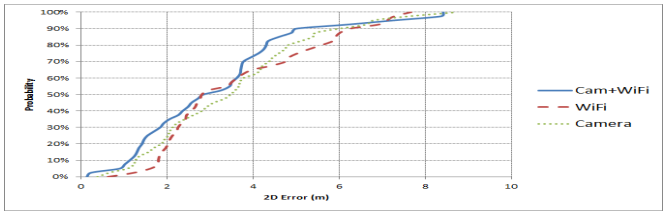
The conditions that were tested included a person with a Wi-Fi client moving around in the test area alone with no interference, this situation is represented by Figure 3(a). Other conditions include a stationary Wi-Fi client while a person walks around, and a cluttered scene where one Wi-Fi client and several others walk in the test area. This is shown in Figure 3(b), displaying no significant increase in accuracy to Wi-Fi, but the location error is seldom worse than Wi-Fi alone.

V. Conclusion

The results indicate that by combining the two measurements the accuracy can be increased while never dipping below the best accuracy the of a single measurement.



(a)



(b)

Fig. 3. **(a)** The cummulative distribution function of the user walking around the test area without interference.**(b)** all other situations

References

[1] R. Zetik, J. Sachs, and R. Thom, "UWB localization - active and passive approach."

[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. Chapter 3, pp. pp. 51–84.

[4] M. Piccardi, "Background subtraction techniques: a review," The ARC Centre of Excellence for Autonomous Systems (CAS) Faculty of Engineering, UTS, Tech. Rep., 2004.

[5] G. Mori, *Particle Filter Notes*, Simon Fraser University, Computing Science, 2005.

[6] 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.

[7] P. Van Hese, S. Gruenwedel, V. Jelaca, J. Nino, and W. Philips, "Evaluation of Background/Foreground Segmentation Methods for multi-view Occupancy Maps," in *Positioning and Context-Awareness Conference*, 2011.

[8] J. A. Bilmes, *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, 1998.

[9] S. Dasgupta, "Learning Mixtures of Gaussians," Computer Science Division (EECS) University of California, Tech. Rep., 1999.