In-plane user positioning indoors

In-plane User Positioning Indoors

Natasa Jovanovic Tanir Ozcelebi Johan Lukkien
Eindhoven University of Technology
P.O.Box 513 Eindhoven, The Netherlands
n.jovanovic@tue.nl t.ozcelebi@tue.nl j.j.lukkien@tue.nl

Abstract

Indoor positioning is a service required by many smart environment applications for various purposes, such as activity classification, indoor navigation and context awareness. In this paper, we present a novel approach to the user positioning problem based on in-plane detection enabled by a set of infrared light emitters and sensors placed horizontally along the walls. The simulation results show that the proposed system is able to determine locations of multiple users inside the room with high precision and accuracy.

1 Introduction

As an enabling service, user positioning is required by many applications ranging from navigation, surveillance and traffic control outdoors, to robot guidance, user tracking and activity recognition indoors. Outdoor positioning, tracking and navigation today are enabled almost exclusively by Global Positioning System (GPS), with every smart phone and car navigation system containing a GPS receiver. However, using GPS for indoor positioning is neither reliable nor accurate as the technology requires line-of-sight with multiple satellites.

To this date, there is no generally accepted indoor positioning system and a multitude of indoor positioning technologies, sensory devices and algorithms that are suitable for different applications have been developed [1]. Most of these utilize cameras [2, 3], passive infrared (PIR) sensors [4, 5] or radio frequency identification (RFID) tags [6, 7]. Each of the sensing techniques and the positioning methods implied by them have unique advantages and limitations. For instance, vision-based positioning systems are criticized mostly because they disturb privacy of people. PIR sensors are widely used because they are cheap, however, there are known issues with the coverage and hidden objects [8], as well as the issues of reporting false detections that are difficult to filter out [9]. The RFID-based positioning systems provide identification of separate users, however, they rely on users carrying a location device (tag) and are therefore considered obtrusive.

User positioning solutions in the literature treat users as point objects and aim to position them with respect to a two dimensional coordinate system. In reality, people occupy an area which may be symmetric, e.g., when they stand straight, or asymmetric, e.g., when they lean. Instead, we formally define the problem of indoor user positioning as follows. Given a regular three-dimensional indoor environment Γ with walls that are perpendicular to the floor and the ceiling, determine the area $\mathcal{D}$ that is the intersection of all users positioned inside Γ and a horizontal two-dimensional virtual detection plane A that cuts Γ at a height of h. Suitable values of h can be different for different applications and indoor settings as explained in Section 2.

In this paper, we explore the possibilities of using in-plane object detection [10] in an indoor environment. The concept of in-plane object detection was developed for positioning and tracking of multiple objects on a two-dimensional rectangular surface. For this, infrared light emitters (LEDs) and sensors are placed at fixed and known positions along the circumference as shown in Figure 1. The LEDs and sensors are

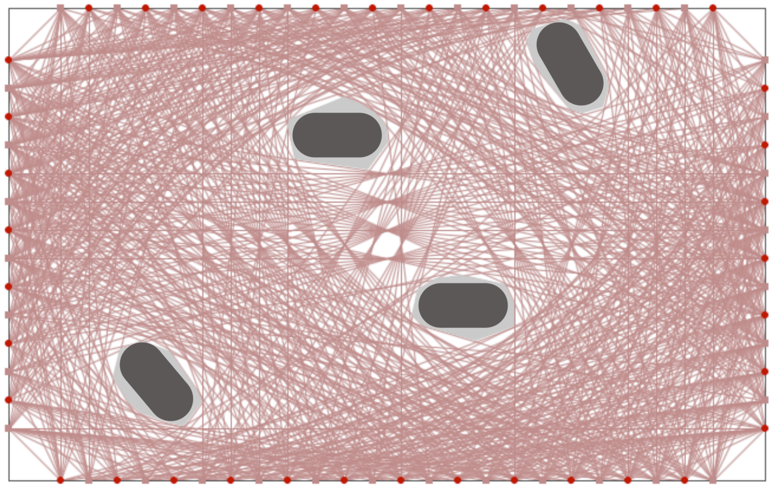

Figure 1: The polygons larger than a given threshold area and that are not intersected by the detection lines are reported as positions.

positioned in an alternating fashion and the distance between any neighboring LED-sensor couple is constant. During a so-called *detection cycle*, the LEDs flash (infrared) light in turns and all sensors report whether or not they sense it each time a LED flashes. The sensor data collected in a detection cycle is saved in the form of a binary matrix where the i-th row corresponds to the i-th LED and the j-th column corresponds to the j-th sensor. If there is an object blocking the light between a LED and a sensor, the corresponding entry in the matrix is set to 0, otherwise the entry is set to 1. This binary matrix denoted as B is called *the blocking matrix* and it is used as an input to the object positioning algorithm. In-plane object detection and positioning became known through implementation in interactive multi-touch screens such as Entertaible [11] and Zero-touch [12], where small objects such as fingers and game pawns are repeatedly positioned, i.e., tracked in real time.

Differently from multi-touch screens, it is not possible to place LEDs and sensors at the entire circumference of an indoor environment, due to doors, windows, cabinets and other objects placed at the circumference. Furthermore, the performance is hampered by other objects detected such as furniture. Therefore, the in-plane detection based object positioning algorithm introduced in [10] cannot be directly applied to user positioning for indoors. In this paper, we introduce a user positioning algorithm that is also based on in-plane detection. We evaluate its performance by measuring *precision* and *accuracy* metrics in four different simulations of indoor environments. Note that repeated use of the proposed user positioning algorithm in combination with a user identification service can be utilized for a more sophisticated algorithm for tracking identified users in real-time. However, this is beyond the scope of this paper and is left as future work. The proposed algorithm simply aims to determine the area occupied by anonymous users at any instance of time.

2 Algorithm for user positioning

The proposed algorithm is based on the following realistic assumptions. The detection plane A is at a height of h from the floor and contains the infrared LEDs and sensors at the circumference. This height can be conveniently chosen during the hardware installation phase, such that the plane intersects the least possible number of furniture pieces, however, it always intersects the users regardless of their standing and sitting positions. For simplicity, we assume that LEDs and sensors are points on the perimeter of the detection plane. For practical reasons, no LEDs and sensors can be placed along the doors, windows and room furniture.

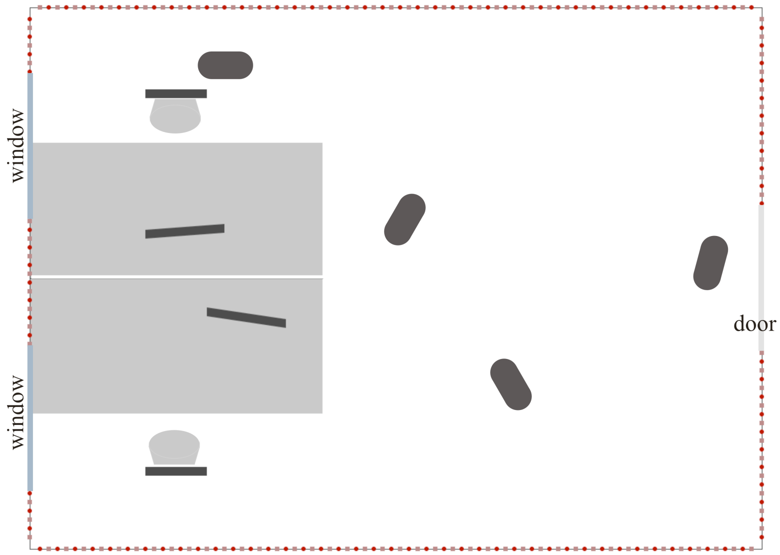

Figure 2: The bird's eye view of the hardware setup on the walls and the interior of an office room: the grey shapes represent the pieces of room furniture that do not intersect the detection plane (because they are either below or above the detection plane), while the black shapes represent four users and room furniture that intersect the detection plane.

The users block the light emitted by the LEDs, therefore, the area occupied by the users in the detection plane cannot be intersected by the so-called detection lines connecting individual LEDs and sensors; see Figure 1. The detection lines that are not blocked (easily determined from the blocking matrix B) divide the detection plane into a large number of convex polygons. The algorithm determines these polygons using a recursive routine of cutting a polygon into two smaller polygons by one of the detection lines intersecting that polygon [10]. After cutting the detection plane with all detection lines, each polygon that is larger than threshold area σ is reported as a position in the output of the positioning algorithm; see Algorithm 1. We choose the value of σ to be equal to the minimum intersection size between a person and the detection plane. Note that the algorithm determines *all* polygons larger than σ, therefore, if there are other large enough objects besides users in the detection plane, these will be reported as well. A polygon is represented by an ordered sequence of its vertices, which are given by their two-dimensional coordinates in A. The polygon representation of the user position represents an improvement over reporting a position in the (x, y)-form, also because it allows the detection of the change in the orientation when the center of the mass does not move, e.g. when turning around in place.

Additionally, to discard the polygons that are larger than σ but may not circumscribe any users, e.g. a very long but thin polygon, it can be checked whether the approximate shape of a user's cross section with A can be inscribed in each polygon. This would, however, introduce high computational complexity to improve on identifying extremely rare occurrences of such polygons.

We consider a user as positioned if the positioning algorithm outputs a polygon circumscribing the user in the detection plane and this instance is marked as a *true positive*. Since the LED-sensor lines blocked by the users are not involved in the cutting of the detection plane, each user by definition must be entirely contained inside a convex polygon formed by the detection lines. This means that the described method cannot miss any user, i.e., not report a user's position, as long as the threshold size is set to be smaller than the smallest cross section (e.g. waist size) of each user. In other words, the algorithm does not give *false negatives*. However, a known intrinsic shortcoming of the in-plane detection is that it can result in *false positives* [13]. A polygon reported by the positioning algorithm is said to be a false positive if it does not circumscribe any users. The false positives occur in situations where all lines intersecting a large enough area are blocked by users (outside the area) and by other objects in the detection plane; see Figure 3.

Algorithm 1 USERPOSITIONING(B)

```
Let 𝒟 denote the set of positions in A;
Determine the set ℒ of all detection lines in A from the blocking matrix B;
procedure DETECT(A, ℒ, σ)
    Determine the set ℒ_cut of lines in ℒ that intersect A;
    if size(ℒ_cut) = 0 then
        if size(P) > σ then
            𝒟 ← A
        end if
        return
    end if
    Choose a line l from ℒ_cut;
    Remove l from ℒ_cut;
    Cut polygon A with l and denote the resulting polygons as D_1 and D_2;
    if size(D_1) > σ then
        DETECT(D_1, ℒ_cut, σ)
    end if
    if size(D_2) > σ then
        DETECT(D_2, ℒ_cut, σ)
    end if
end procedure
```

The performance of the algorithm is measured using precision and accuracy metrics. We define the *precision* as the ratio between the number pos_{true} of true positives and the sum of the numbers pos_{true} and pos_{false} of the true positives and the false positives, respectively, i.e.,

$$precision = \frac{pos_{true}}{pos_{true} + pos_{false}} \tag{1}$$

In addition, *accuracy* is a metric that describes how well the reported positions correspond to the actual area that users occupy in the detection plane. More precisely, if the areas that N users occupy are $U_1, U_2, \ldots, U_N$ and the areas of each of the K positions reported in $\mathcal{D}$ are $S_1, S_2, \ldots, S_K$, the accuracy of the user positioning is given by

$$accuracy = \frac{U_1 + U_2 + \cdots + U_n}{S_1 + S_2 + \cdots + S_k} \tag{2}$$

While higher precision implies a lower rate of false positives, higher accuracy implies a tighter fit between the area determined by the positioning algorithm and the actual area of intersection with users. The described positioning algorithm relies on the presence of a large number of detection lines connecting the LEDs and sensors. Naturally, the performance would improve with respect to precision and accuracy with increasing the number of detection lines, i.e., by increasing the number of LEDs and sensors that define them. However, regardless of the density of LEDs and sensors, it can be that multiple users are reported as a single polygon and some of the polygons that are reported may not circumscribe users (false positives). In other words, the number of users N can be different from the number of positions reported K. Note that detecting multiple users in a single polygon instead of separate polygons affects accuracy, but it does not affect precision. This is because the area occupied by the users is still reported, although jointly within a single polygon.

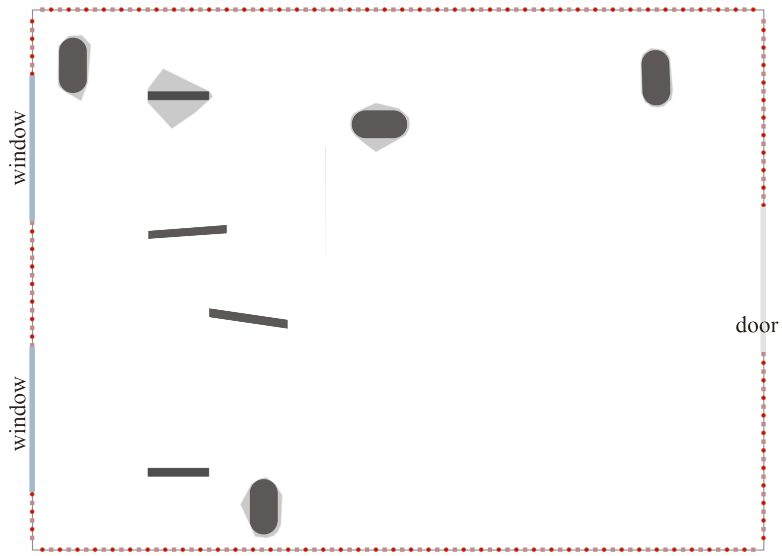

Figure 3: Four users and four pieces of furniture result in five positions reported; four of these are true positives and one is a false positive.

3 Simulation results

In order to investigate the applicability of the in-plane detection method for user positioning and evaluate the proposed user positioning algorithm, we developed a simulation platform. The simulation platform incorporates three major components that define the simulation environment and that can be changed independently. These components are:

- hardware configuration, defining all parameters related to the LEDs and sensors, such as their count and relative positions;
- room layout, defining all objects that can be present in the room such as the furniture;
- user model, defining all parameters related to the users, i.e. the number of users, user cross section size, shape, position and orientation.

We measured the precision and the accuracy of the positioning algorithm as defined in Equations (1) and (2) in four environments, the result of combining two hardware configurations and two room layouts for an office space of length 500 cm and width 370 cm. The detailed description of these four environments is presented below.

Environment 1. This environment is created to explore the precision of user positioning under overly optimistic conditions. The LEDs and sensors are uniformly

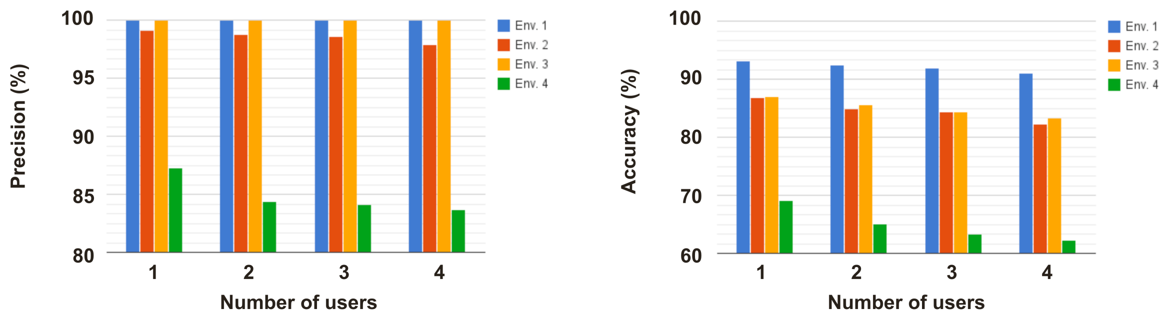

Figure 4: The average precision (left) and accuracy (right) of in-plane detection determined for 1000 different user locations.

placed along the entire circumference on the detection plane in an alternating fashion, including the walls, all windows and doors. There are 142 LEDs and 142 sensors in total, and the distance between a LED and a neighboring sensor is 6 cm. The room contains no furniture that intersects the detection plane, in other words, the detection plane is considered to be empty when there are no users in the room.

Environment 2. This environment has the same hardware configuration as Environment 1. However, to create a realistic model of an office, we place two desks, two chairs and two monitors in the room layout. We assume that the height of the detection plane is such that it does not intersect the desks. Hence, the desks cannot block the infrared light emitted by LEDs, however, their presence restricts the movement of users inside the room. The intersection of a chair and the detection plane, as well as the intersection of a monitor and the detection plane are modeled as a rectangle of size 35 cm by 6 cm and a rectangle of size 50 cm by 6 cm, respectively.

Environment 3. The hardware configuration of this environment assumes there are no LEDs and sensors placed along the two windows of width 100 cm and one door of the same width. The windows are assumed to be on the wall opposite to the wall containing the door. There are 118 LEDs and 118 sensors in total, and the distance between a LED and its neighboring sensor is 6 cm. The room is assumed to be empty, as it is in Environment 1.

Environment 4. This environment represents a realistic model of a small office, with the hardware configuration as in Environment 3 and the room layout as in Environment 2.

For simplicity, a user's cross section with the detection plane is modeled to have the shape of the convex hull of two disks of radii 10 cm that are tangent to each other from the outside. For each separate case of 1, 2, 3 or 4 users present in one of the four defined environments we measured the precision and the accuracy of the in-plane positioning over 1000 tests, where each test corresponds to a new random set of positions that users occupy. The results are presented in Figure 4 and Table 1.

Table 1: The average precision and accuracy of in-plane detection determined for 1000 different user locations.

	Environment 1		Environment 2		Environment 3		Environment 4	
#Users	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.
1	100	93.1	99.1	86.9	100	87.1	87.3	69.1
2	100	92.5	98.8	84.9	100	85.6	84.4	65.1
3	100	91.9	98.6	84.3	100	84.4	84.1	63.3
4	100	91.1	97.9	82.3	100	83.3	83.7	62.2

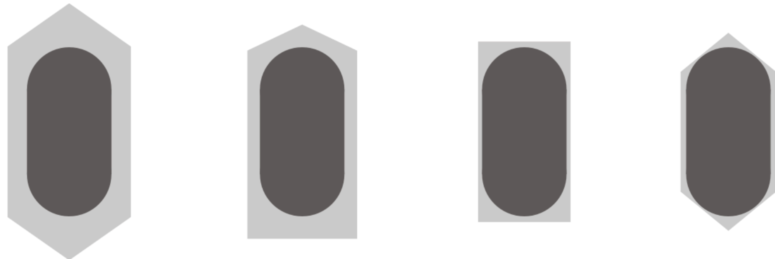

Figure 5: Positioning samples with various accuracies.

With the minimum precision of 83.7% and the minimum accuracy of 62.2%, the in-plane positioning method shows very good performance in multiple users positioning in a relatively small office space. Figure 5 shows positioning instances and the corresponding (individual) accuracies.

4 Conclusion and future work

We proposed an algorithm that utilizes in-plane object detection for user positioning in indoor environments. This anonymous and unobtrusive technology is enabled by a set of infrared LEDs and sensors placed in an alternating fashion on the walls of the given indoor space. The proposed algorithm reports the positions in the form of convex polygons circumscribing users within a horizontal detection plane. Using this algorithm, we simulated user positioning in four environments and measured the precision and the accuracy of the positioning method for up to four users.

The precision of 100% and the accuracy of more than 83% in Environments 1 and 3 clearly indicate the direction that should be taken to improve the positioning precision in reality. The relatively large gaps in the LED-sensor frame practically do not affect the positioning method. In contrast, large obstructing objects in detection plane, as the ones in Environments 2 and 4, can cause false positives, usually those same objects being reported as user positions. Therefore, in order to minimize the risk of false positives, the height of the detection plane should be chosen such that it has minimum intersection with objects in the room. In addition, in future work, false positives may be identified by comparing positions reported in consecutive detection cycles and removing those that appear and disappear in the middle of the detection plane causing a kind of blinking polygons effect. Alternatively, multiple detection planes on different heights can be deployed to eliminate the false positives by combining results and to ensure the highest precision and accuracy of user positioning.

References

[1] Y. Gu, A. Lo and I. Niemegeers, *A Survey of Indoor Positioning Systems for Wireless Personal Networks.* In IEEE Communications Surveys & Tutorials, vol. 11, no.1, 2009.

[2] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, *Multi-camera multi-person tracking for easy living.* In Proc. of 3rd IEEE Int'l Workshop Visual Surveillance, 2000.

[3] D. Focken and R. Stielfelhagen, *Towards vision-based 3–D people tracking in a smart room.* In Proc. of 4th IEEE Int'l Conf. on Multimodal Interfaces, 2002.

[4] A. Harter and A. Hopper, *A distributed location system for the active office*. IEEE Network, vol.8, no.1, pp. 62–70, 1994.

[5] E. Aitenbichler and M. Mhlhuser, *An IR local positioning system for smart items and devices*. In Proc. of 23rd IEEE Int'l Conf. on Distributed Computing Systems Workshops, 2003.

[6] L. M. Ni and Y. Liu, *LANDMARC: Indoor location sensing using active RFID*. In Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications, pp. 407–416, 2003.

[7] H. D. Chon, S. Jun, H. Jung and S. W. An, *Using RFID for accurate positioning*. In Proc. of Int'l Symposium on GNSS, Sydney, 2004.

[8] R. Casas, D. Cuartielles, A. Marco, H. J. Gracia and J. L. Falc, *Hidden issues in deploying an indoor location system*. In IEEE Pervasive Computing, vol.6, no.2, pp. 62–69, 2007.

[9] X. Fernando, S. Krishnan, H. Sun and K. Kazemi–Moud, *Adaptive denoising at infrared wireless receivers*. In Proc. of SPIE, 2003.

[10] N. Jovanović, J. Korst and V. Pronk, *Object detection in Flatland*. In Proc. of the 3rd Int'l. Conf. on Advanced Engineering Computing and Applications in Sciences, Malta, 2009.

[11] G. Hollemans, T. Bergman, V. Buil, K. van Gelder, M. Groten, J. Hoonhout, T. Lashina, E. van Loenen and S. van de Wijdeven, *Entertaible: Multi-user multi-object concurrent input*. Adjunct proc. of UIST 6, pp. 55–56, 2006.

[12] J. Moeller and A. Kerne, *ZeroTouch: An optical multi-touch and free-air interaction method*. In Proc. of CHI 2012, pp. 2165–2174, 2012.

[13] N. Jovanovic, J. Korst, Z. Aleksovski, W. Michiels, J. Lukkien, E. Aarts, *Finding shadows among disks*. In Proc. of the 24th Canadian Conference on Computational Geometry, pp. 89–94, 2012.