Marker-Free Indoor Localization and Tracking of Multiple Users in Smart Environments Using a Camera-Based Approach

Andreas Braun[1], Tim Dutz[1], Michael Alekseew[2], Philipp Schillinger[2], and Alexander Marinc[1]

[1] Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
{andreas.braun,tim.dutz,alexander.marinc}@igd.fraunhofer.de
[2] Technische Universität Darmstadt, Darmstadt, Germany
{michael.alekseew,philipp.christian.schillinger}@stud.tu-darmstadt.de

Abstract. In recent years, various indoor tracking and localization approaches for usage in conjunction with Pervasive Computing systems have been proposed. In a nutshell, three categories of localization methods can be identified, namely active marker-based solutions, passive marker-based solutions, and marker-free solutions. Both active and passive marker-based solutions require a person to carry some type of tagging item in order to function, which, for a multitude of reasons, makes them less favorable than marker-free solutions, which are capable of localizing persons without additional accessories. In this work, we present a marker-free, camera-based approach for use in typical indoor environments that has been designed for reliability and cost-effectiveness. We were able to successfully evaluate the system with two persons and initial tests promise the potential to increase the number of users that can be simultaneously tracked even further.

Keywords: Indoor localization, Computer Vision, Pervasive Computing.

1 Introduction

Reliably localizing and tracking multiple users in smart environments has evolved into one of the main challenges of this research area. The knowledge of the users' whereabouts is a central contextual information to an assistive system and oftentimes plays a pivotal role when such a system needs to decide, whether it is supposed to act; that is, whether it should influence the current state of its environment through its actuators. And although simple motion sensors can be used to provide for basic presence detection, much more sophisticated solutions are required for the concurrent localization of multiple people within the same area – or simply to distinguish between a person's pet, and herself.

In recent years, various indoor tracking and localization approaches for usage in conjunction with Ambient Intelligence systems have been proposed and there are even specific competitions with the intention of comparing the different methods'

N. Streitz and C. Stephanidis (Eds.): DAPI/HCII 2013, LNCS 8028, pp. 349–357, 2013.

performances against one another [1]. We can distinguish three different categories of localization methods, namely active marker-based solutions, passive marker-based solutions, and marker-free solutions. Both active and passive marker-based solutions require a person to carry some type of tagging item in order to function, which, for a multitude of reasons, makes them less favorable than marker-free solutions, which are capable of localizing persons independently of whether they are carrying additional accessories. Examples for approaches from this latter category include capacitive sensitive floors [2], using microphones for the detection of subtle noises caused by movement [3], and camera-based approaches [4]. The three main criteria that all of these localization solutions are judged on are the total costs for providing them for a specific area, such as a private apartment, their reliability, and the amount of persons that can be tracked and distinguished by them at a time.

In this work, we present a marker-free, camera-based approach for use in typical indoor environments, which allows the reliable localization of multiple persons. The system tested is able to successfully track two users in parallel.

2 Related Work

Detecting the presence and location of persons has been a research effort for many decades and as such, can now be achieved using a variety of technologies. Capacitive sensors use oscillating electric fields to measure the properties of an electric field, allowing the presence of a human body to be detected. Braun et al. have presented a system using electrodes laid out in a grid and hidden under floor covering to detect the location of one or more persons [2]. A similar system that integrates necessary electronics into a floor layer and communicates wirelessly to a central system has been presented by Lauterbach et al. [5]. Both systems furthermore allow the realization of additional use cases, such as intrusion detection and fall prevention.

Walking is creating a certain level of noise that can be picked up by microphones and used to infer the location of persons. Most of these systems use time-of-flight techniques; that is, calculating the distance of the source by measuring the time required for the signal to arrive at a specific location and triangulating its position [6]. While earlier system relied on speech to recognize sound sources [3], newer and more sensitive systems allow the detection of a person from the sound of the person's footsteps [7].

Another popular method is based on different radio frequency techniques, e.g., by measuring signal strength (RSSI) on different receivers and triangulate position [8] that require an active token to be worn. A newer approach is using tomography techniques to measure the signal attenuation by human bodies [9] and allows localization without wearing active tokens.

Finally, the method that our work is based on comes from the area of computer vision and uses different types of cameras [10], depending on visible light or infrared depth imaging [11]. Most systems use similar approaches that use background subtraction to detect movement in single images or time-series of images to infer the position of an object [4].

3 System Design

In this section, we describe the rationale that has been driving the development of our system with the specific requirements important in cost-effective personal localization solutions and how it has affected both the hardware architecture and software behind the system.

3.1 System Requirements

The system is based on a set of standard, off-the-shelf webcams and as such excels through its low cost factor – the hardware cost for an average living room should be less than fifty US$ (provided that a PC is already available). There are three main challenges associated to the design of such a system for smart environments:

- Scalability - it should be easy to attach additional cameras to the system and provide tools that allow setting position and orientation of the video devices within the environment
- Computational Feasibility - the algorithms used for person localization should be suitable for usage with low-resolution, low-bandwidth data, while still being able to reliably recognize moving persons
- Flexibility - the system should be able to distinguish between different persons and discard other moving objects, such as pets

The system we are using is set up using a simple configuration tool that models the environment and the extrinsic camera parameters by way of XML files. The video stream of each camera is analyzed for signs of movement and we register the results of each camera to the others. This allows the generation of three-dimensional data of moving objects and the inference of such an object's position within the environment. At least two cameras must capture the moving object for the method to work. In border cases, we use approximations and historical movement data to estimate the object's position. We are using simple metrics to distinguish between different persons, based on the color of their clothing and body volume.

3.2 Hardware Architecture

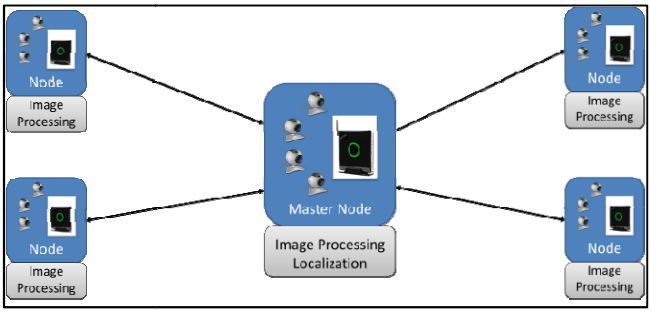

Fig. 1. Hardware architecture of the localization system

The system is comprised of various nodes made up of a single PC with various USB cameras attached. They are connected to each other using either a wired LAN connection (preferably), or WiFi. The cameras used should be controllable in terms of modifying their settings, such as automatic settings of white balance, gain, contrast and brightness. This allows offloading the image processing to the single systems and in consequence only higher-level features are sent through the network connections. This is reducing the required bandwidth and making this approach feasible for low-speed wireless networks. One of the nodes is acting as a master, analyzes the high-level data and provides the overall processing of the final localization. The overall architecture is shown in .

3.3 Data Processing

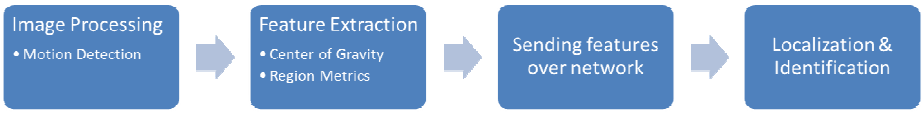

Fig. 2. Localization process

Our system is following a regular camera-based indoor localization process, as shown in **Fig.2**. Each individual system is processing the image of the camera using a motion detection algorithm. We use a custom variant of background subtraction that allows a fine-grained control of sample window, camera parameters and feature size, guaranteeing swift adaptation to different room geometries. In a second step, we extract features from the detected motion, in our case the center of gravity of each moving region and metrics about the detected regions, that allow us to identify persons in a future iteration. Only these features (and not the entire stream) are then sent over the network for further processing. Finally the master system is collecting all the features, combines it with its local representation of the environment and performs localization of the different persons.

A control software has been created that realizes all these steps. Furthermore, it provides various tools supporting this process. These tools include:

- Camera management: add/remove cameras, set intrinsic and extrinsic parameters
- Environment management: read layout from image files
- Camera placement tools: coverage analysis, coverage optimization
- Performance analysis: show CPU load, network status, logging

Fig.3. shows a screenshot of the software's main user interface. On the left, we can see the source image file of the environment. Using a threshold-based processing, the boundaries are extracted from the black areas indicating walls. The environment can be extracted from any layout file that uses similar dark areas for boundaries. On the right side of the figure, we can see the wizard that allows the adding of additional cameras.

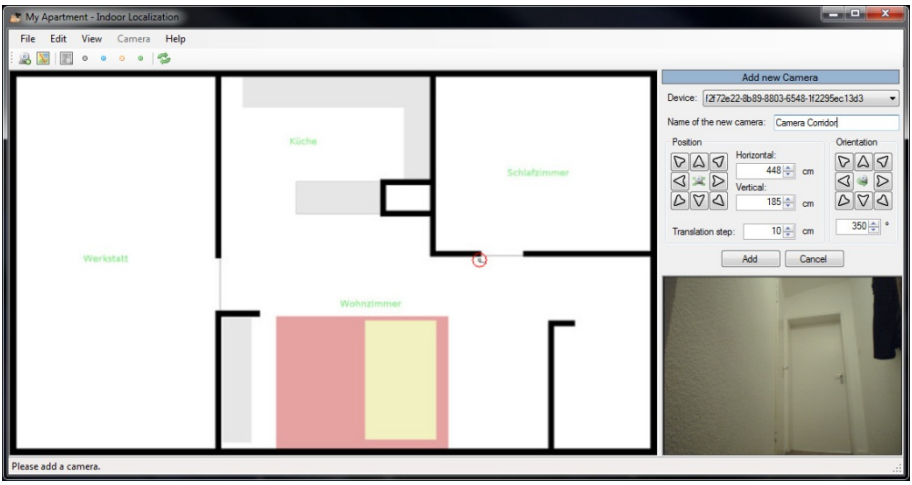

Fig. 3. Software's main view

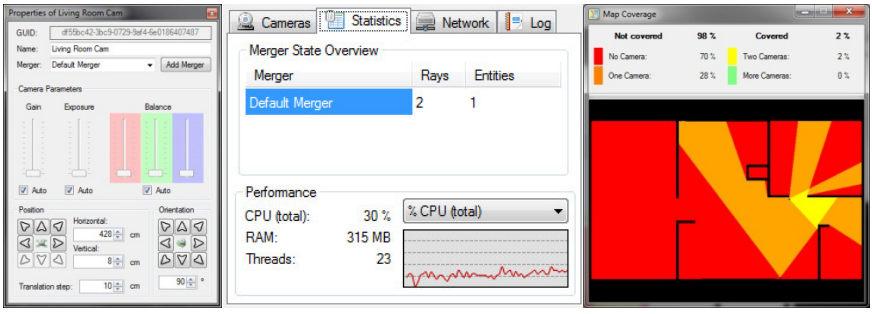

Fig. 4. Camera properties (left), Statistics (center), Coverage analysis (right)

The wizards of the software enable us to modify the position and orientation of cameras and check on the live camera stream. Once a camera is added, it is also possible to control the results of the image processing in a dedicated window and individually set post-processing parameters, such as white balance and color correction (**Fig.4.** - left). The statistics window as shown in **Fig.4** (center) gives an overview of available master nodes (mergers) and the load on all available CPU cores as well as the number of currently active threads. Finally, the coverage map shown **Fig.4** (right) displays by color, which areas of the environment are currently covered by cameras, and by how many (red indicates blind spots, orange areas are in the view of a single camera, yellow areas are surveyed by two cameras, and orange areas are covered by at least three cameras). We have found that, as a rule of thumb, a reliable localization is achieved for all yellow and green areas (areas covered by at least two cameras).

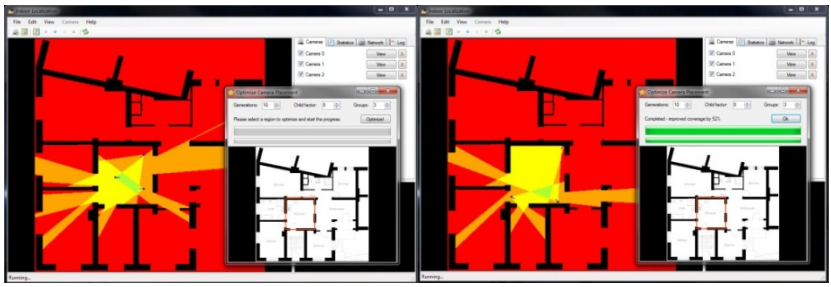

Fig. 5. Before and after camera placement optimization

An interesting feature of the software related to these areas of coverage is an optimization algorithm for camera placement. Using the camera coverage area as a quality metric, a genetic algorithm is used to calculate optimal camera positioning. The algorithm is optimizing camera placement based on the number of available cameras and is considering wall and ceiling positions as an additional restriction.

The software was created using C# and the .NET runtime environment. For image processing, we are using EmguCV[1], a .NET wrapper for OpenCV[2]. This is a comprehensive image processing and computer vision library, which already provides many of the methods required.

4 Prototype

Fig. 6. Playstation Eye camera out-of-the-box (left) and hanging upside-down in the custom-built stand (right)

[1] http://www.emgu.com
[2] http://opencv.org/

For our prototype set-up, we selected the Playstation Eye as the camera of our choice, because it is available at a low price and nevertheless allows the setting of parameters such as frames per second (FPS), deactivation of auto-white-balance and auto-contrast, as well as setting exposure and gain. Manually controlling these parameters is crucial in image processing applications and not a general feature available for all cheaper variety web cams. Additionally, we have designed a custom stand that allows the easy attachment of the cameras on walls, as shown **Fig.6**. While the system is easily scalable, for our initial tests we have used only two nodes, with two cameras attached to each of those (which results in a total of four cameras). This setup has proven to be sufficient for covering a large room (roughly 35 square meters). Both nodes were running our software on 64-bit multi-core processors (AMD Turion 64 and Intel Core i5). The cameras are running at 30 FPS and VGA resolution (640x480). The CPU load and amounts of threads used indicate that each node would be able to handle at least twice the number of cameras. RAM requirements have generally shown to be fairly low.

The system was installed in our institute's Living Lab, which consists of a combined living room and kitchen area, a bedroom and an office. For our evaluation, only the combined living room and kitchen area were considered. The area covered is approximately 35 square meters and is occupied by several large pieces of furniture (cupboards, desks, and the like). Therefore, a sophisticated camera placement is crucial in order to guarantee good coverage. The software tools as described were essential for finding optimal camera positions in this setting. As a next step, we intend to extend our prototype to all rooms of the Living Lab.

5 Evaluation

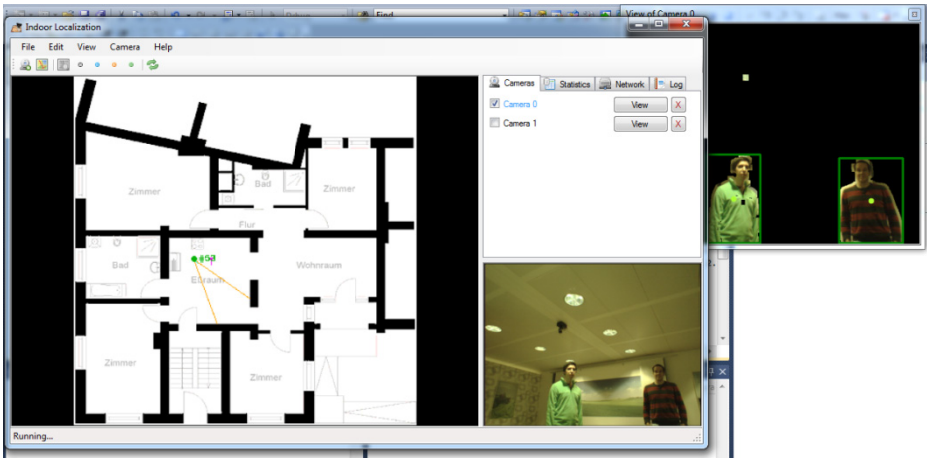

Fig. 7. The simultaneous detection of two persons

As indicated before, we have been able to successfully test our prototype system (software and hardware setup) for the simultaneous tracking of two persons, using four cameras to cover an average-sized living room. By using the camera placement optimization algorithm, we positioned the four cameras on different corners of the room and thus maximized the area covered by at least three devices. The screenshot one can see on Fig. 7 shows the software's main screen with the apartment's map on the left and one camera's viewing angle highlighted. The image stream of this selected camera can be seen on the lower right. The frame on the upper right shows the persons that are currently tracked by this camera (supported by the feeds delivered by the other three cameras). Making use of small markers on the floor, we have been able to verify that our system's distance estimation feature is already fairly precise for areas that are covered by at least three cameras (the estimation was rarely more than 30 cm off the mark). The processing power of the two medium-class PCs we used for handling the four camera's image streams proved to be more than capable for this and showed significant reserves. Based on this, we intend to build a new prototype system which will use only a single PC for handling four, six or maybe even eight cameras at once and which could then be used to monitor an entire small apartment.

6 Conclusion and Future Work

In this work, we have presented a system for the indoor localization of multiple persons. Based entirely on affordable hardware and open source software libraries, we created a reliable, scalable and versatile solution for tracking up to two users within a large indoor area. The hardware costs for the system itself are approximately $100 for four cameras, stands and cabling. Because multiple cameras can be attached to a single computer (mainly depending on its processing power), the overall costs of the system are not dependent on the number of cameras used. So far, we have been able to use our system for the tracking of two users in our equipped demonstration area. Additionally, we have integrated innovative aspects in the processing pipeline, such as camera coverage optimization using genetic algorithms and network analysis.

Nonetheless, the prototype as presented in this work is merely an intermediate step. As future work, we intend to scale up the system to be able to cover entire apartments and multiple separated rooms. Also, the identification of specific users as realized in its current state is rudimentary at best and requires further testing. As a next step, we will thus test different identification features and investigate, how many persons we can easily differentiate. In terms of hardware we would like to evaluate different types of cameras, such as the Microsoft Kinect for depth imaging, which allows a more reliable background subtraction and thus is potentially better suited for scenarios where many users are present. Finally, we would also like to test self-organizing networks for smart cameras that perform image processing on an included chip and send features to each other using wireless communication systems. Self-localization and registration are further aspects we would like to explore in this regard.

References

1. Chessa, S., Knauth, S.: Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking. Springer, Heidelberg (2012)
2. Braun, A., Heggen, H., Wichert, R.: CapFloor – A Flexible Capacitive Indoor Localization System. In: Chessa, S., Knauth, S. (eds.) EvAAL 2011. CCIS, vol. 309, pp. 26–35. Springer, Heidelberg (2012)
3. Sturim, D.E., Brandstein, M.S., Silverman, H.F.: Tracking multiple talkers using microphone-array measurements. IEEE Comput. Soc. Press (1997)
4. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for EasyLiving. In: Proceedings Third IEEE International Workshop on Visual Surveillance, pp. 3–10. IEEE Comput. Soc. (2000)
5. Lauterbach, C., Steinhage, A.: SensFloor ® - A Large-area Sensor System Based on Printed Textiles Printed Electronics. In: Ambient Assisted Living Congress. VDE Verlag (2009)
6. Brandstein, M.S., Silverman, H.F.: A practical methodology for speech source localization with microphone arrays. Computer Speech Language 11, 91–126 (1997)
7. Guo, Y., Hazas, M.: Localising speech, footsteps and other sounds using resource-constrained devices. In: 10th International Conference on Information Processing in Sensor Networks (IPSN), pp. 330–341 (2011)
8. Balakrishnan, H.: The Cricket Indoor Location System. Doctoral Dissertation, Massachusetts Institute of Technology (2005)
9. Wilson, J., Patwari, N.: See-Through Walls: Motion Tracking Using Variance-Based Radio Tomography Networks. IEEE Transactions on Mobile Computing 10, 612–621 (2011)
10. Lopez de Ipina, D., Mendonça, P.R.S., Hopper, A.: TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing. Personal and Ubiquitous Computing 6, 206–219 (2002)
11. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR 2011, pp. 1297–1304 (2011)