# Using SIFT and WiFi Signals to Provide Location-Based Services for Smartphones

Antonio J. Ruiz-Ruiz, Oscar Canovas, Ruben A. Rubio Muñoz, and Pedro E. Lopez-de-Teruel Alcolea

Department of Computer Engineering, University of Murcia, Spain
{antonioruiz,ocanovas,rrm15947,pedroe}@um.es

**Abstract.** This paper introduces how to fuse the data acquired from different sensors available in commodity smartphones to build accurate location-based services, pursuing a good balance between accuracy and performance. Using scale invariant features from the images captured using the smartphone camera, we perform a matching process against previously obtained images to determine the current location of the device. Several refinements are introduced to improve the performance and the scalability of our proposal. Location fingerprinting, based on IEEE 802.11, will be used to determine a cluster of physical points, or zone, where the device seems to be according to the received signal strength. In this way, we will reduce the number of images to analyze to those contained in the tentative zone. Additionally, accelerometers will be also considered in order to improve the system performance, by means of a motion estimator. This set of techniques enables a wide range of new location-based applications.

**Keywords:** SIFT, image processing, 802.11, probabilistic techniques.

## 1 Introduction

Location technologies provide a way of associating information to a user position. For a diverse set of areas including tracking, geographic routing or entertainment, location-sensing systems have been an active research field. We will focus on indoor environments, where the Global Positioning System (GPS) suffers from several obstacles blocking the radio signals.

In recent years, the wide adoption of smartphones equipped with several sensors has simplified the process of obtaining the required context information and provides an exceptional starting point to develop location-based services, like augmented-reality applications [10]. Localization systems might benefit from the images that users obtain from the camera in order to display the augmented reality. Indeed, those images can be used to provide a better estimation of the users' position, offering a seamless integration of the sensor and the display. There are several augmented-reality applications which do not require a real time response, such as those designed to display location-aware reminders or notes [19], or to obtain information about who is behind the door of a particular room or laboratory. We find our approach a valuable contribution for this kind of

A. Puiatti et al. (Eds.): MobiQuitous 2011, LNICST 104, pp. 37–48, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

location-based services, since we have obtained the required tradeoff between a fine-grained accuracy and an acceptable response time using data from multiple sensors.

In our proposal, we make use of the scale invariant feature transform (SIFT) [14], an image processing technique suitable for object recognition, to deal with the images captured by the smartphone camera. Our system looks for matching features between the current image and a geo-tagged database of features that were extracted using representative images of the application environment.

Despite we use the images obtained from the camera as the main piece of data to infer the position of a particular device, we can benefit from other sensors in order to limit the amount of information to be examined. Most of the well-known proposals for indoor positioning are based on the received signal strength intensity (RSSI) of transmitted 802.11 packets. In fact, several works have demonstrated that significant accuracy can be obtained in indoor scenarios by means of fingerprinting techniques [7]. As we will show, using fingerprinting methods, we can obtain a cluster of physical points where there is a high probability of locating the device. This cluster is then used to reduce the amount of images in the database to check against, since only those images contained in the selected cluster are analyzed. Additionally, the use of a RSSI-based fingerprinting technique is an added-value service for our purposes, since we will be able to calculate a coarse-grained location estimation which can be useful in those situations when the images obtained by the camera are useless.

Our system achieves acceptable performance distributing the required functionality between the mobile devices and dedicated servers. Additionally, in-built accelerometers will be used to infer coarse-grained user-motion, so reducing the required amount of operations to perform in some circumstances. The system can currently complete an entire cycle, from obtaining data from sensors to providing a location estimation, in less than 3 seconds.

## 2 Related Work

Indoor positioning is a research field that has been addressed by several authors and disciplines. Several types of signals (radio, images, sound) and methods have been used to infer location. Each method has specific requirements as to what types of measurements are needed. Most of the pattern recognition methods, like fingerprinting [7], estimate locations by recognizing position-related patterns. The analysis of RSSI patterns is a technique that has been examined by several authors [4,8], obtaining an accuracy ranging from 0.5 to 3 meters.

Better results can be obtained by integrating the information captured by multiple sensors. SurroundSense [3] is an interesting mobile phone-based location system for indoor environments via ambience fingerprinting of optical, acoustic and motion attributes. However, the optical recognition techniques proposed in SurroundSense are too limited, since the authors are only considering pictures of the floor in order to extract information about light and color. With SIFT, our work avoids typical variations related to light, color or scale, and provides robustness and better accuracy.

In [1,22], the authors performed a preliminary analysis of how techniques like SIFT can be used to improve the accuracy in location based systems. However, both works fail to show an exhaustive analysis of the enhanced possibilities of such techniques when combined with the additional sensors commonly available on smartphones.

Miyaki et al. [15] described an object tracking system for outdoor environments fusing information from CCTV cameras and WiFi signals using a particle filter. While this proposal is based on object tracking (especially people), our work is focused on object recognition in the scene captured by the smartphone camera carried by the user.

Other works also use WiFi signals and image recognition to estimate positions, such as [9] and [17]. However, they are based on two-dimensional landmarks that must be placed in the scenario of interest, which involves the inclusion of obtrusive elements. As we will show, we can obtain good accuracy without imposing such requirements.

## 3 Experimental Environment

The testbed where our experiments were conducted is located on the third floor of our Faculty where several students, researchers and professors move around constantly. The dimension of the testbed is 35 meters by 30 meters, and includes 26 rooms. We have defined a discrete space model of 94 cells where we can link location-based information.

To deploy the fingerprinting system based on RSSI, we distributed six 802.11 access points throughout our dependencies (red dots in Figure 1). Access points already deployed in the scenario might also be used since there are no specific requirements imposed. During the corresponding training phase we collected 250 RSSI observations at each cell. For our database of images, we have obtained a set of seed images, where the number of images captured at each cell depends on the cell type (zoom on Figure 1). Thus, a cell in a corridor is associated to as many images as possible directions of movement, whereas inside the rooms we build panoramic images covering the whole dependency. For larger scenarios we plan to develop some technique like the one presented by Park et al. in [18] in

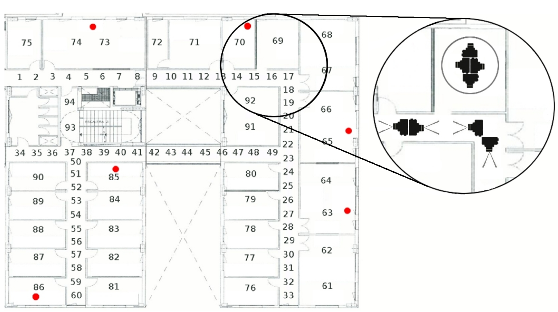


**Fig. 1.** Experimental environment

order to populate larger databases of images and RSSIs using the data provided by the users as they make use of the available location services.

Our experiments were carried out using several hardware devices. The training observations were captured with an Asus Eee 1201 laptop with a Realtek TRL8191SE Wireless LAN 802.11n card. In addition, during the online phase we have also used HTC Desire and HTC Legend smartphones with Android. We developed the appropriate software client for each device in order to collect RSSIs and images and to send them to a repository. Applications were programmed in C++ and Java, depending on the requirements imposed by each device. We have used Linksys WRT54G access points with 802.11abg support. Their locations were chosen so as to provide consistent coverage throughout the entire scenario, guaranteeing that every cell is covered by, at least, 3 access points.

For image processing we use the SIFTGPU library, a GPU Implementation of Scale Invariant Feature Transform (SIFT), implemented by Changchang Wu. SiftGPU requires a high-end GPU, that makes it impossible to be managed by smartphones, so we use a nVidia GeForce 9800GT supporting both GLSL (OpenGL Shading Language) and CUDA (Compute Unified Device Architecture). This GPU has been installed in a Intel(R) Pentium(R) Dual-Core CPU E2160 server. This computer is responsible for estimating the location of the different devices. The resolution of the training images is 640x480 pixels for corridors and 2900x360 pixels for panoramic images of offices and laboratories, and they were captured using a HTC Desire smartphone.

The system has been tested by five different users. They held the phone out in front of them, facing ahead in order to obtain location-aware information related to their current cell or to a zone containing the cell. We are aware that the use of these augmented-reality applications based on images would be sporadic, since we do not envision realistic scenarios where users are comfortable holding their phones facing ahead all the time. Therefore, when the smartphone is inside the pocket, facing the floor, or obtaining useless images (unfocused, uniform), the location estimation will be based mainly on the RSSI measurements.

## 4 Clustering Based on RSSI

As mentioned, several research works have demonstrated that a significant accuracy can be obtained by means of location fingerprinting based on RSSI. We have performed several tests using different techniques in order to compare their results [4,8]. In Figure 2 we compare the accuracy provided by the different techniques that we analyzed. These experiments have been carried out using the training observations as inputs for our estimator and, therefore, we are aware that these results can only be obtained under ideal conditions. We decided to represent the position as a probability distribution using Bayesian inference. As we can see, a histogram-based representation of the sensor model performs better in our scenario.

Accuracy is considerably enhanced by introducing a Hidden Markov Model (HMM) [11]. We have designed the HMM chain as a matrix of NxN size, where N is the number of cells within our scenario. Our matrix has been initialized

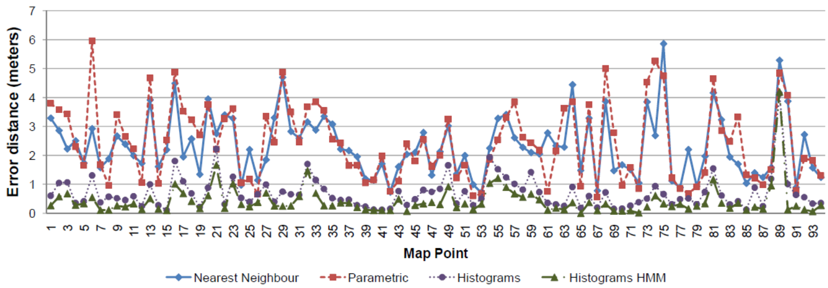



**Fig. 2.** Estimation error using RSSI

using the adjacency relationship between cells and considering that users do not usually move faster than 2 meters per second.

One of the most interesting conclusions that can be obtained from the fingerprinting methods is their suitability to define geographical areas where signals show a similar behavior. Lemelson et al. [12] proposed the Fingerprint Clustering algorithm, which makes use of the training RSSIs to find clusters. It is based on the idea that the signals collected in nearby cells tends to cover only a limited range of the possible values. In order to show how this proposal can be applied in our interests, we have calculated the clusters, shown in Figure 3 (cells pertaining to the same cluster are displayed using the same color), and then we performed several tests where most of the already-analyzed techniques obtain a high cluster hit percentage, up to 93%. According to these experiments, we have defined four overlapping zones joining adjacent clusters in order to reduce the search space when other sensors are also analyzed.

As we will see, the main drawback of using images is the elevated computational cost, especially in huge scenarios, where the number of images to analyze is excessive, involving serious scalability problems. The use of the mentioned clustering technique and the defined zones reduces the number of processed images to those contained in a specific zone of the entire scenario, so performing a fine-grained localization and improving the system scalability.

## 5 SIFT Analysis

The Scale Invariant Feature Transform [13,14] is a widely adopted technique in computer vision research. It provides a method for extracting a collection of features from images, and these features are invariant to image translation, scaling and rotation, and partially invariant to illumination changes and affine distortion or change in 3D camera viewpoint. These features are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. One of the most important characteristics of SIFT is that each feature vector is highly distinctive, which allows a single feature to be correctly matched with a high probability against a large database of features, providing a basis for object and scene recognition.


**Fig. 3.** Clusters distribution

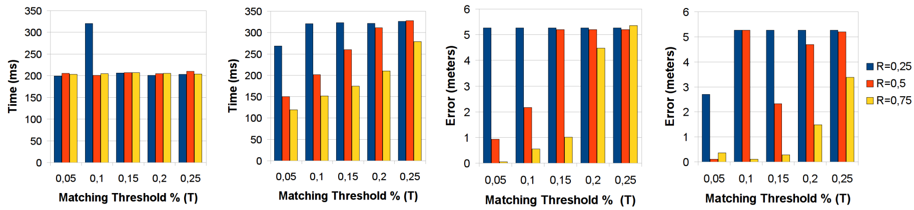
There are several efficient algorithms to extract distinctiveness feature vectors from images, proposed by Lowe [14] or by Vedaldi [21]. Turcot and Lowe have taken an additional step in [20], improving performance by selecting only a small subset of the training features.

Since the SIFT extractor algorithm is based on a Gaussian pyramid of the input image, there are several parameters that influence its performance. We have empirically analyzed the SIFT extractor algorithm in order to find the input parameters that best fit our images and scenario characteristics. We tested different values for the number of octaves, for the index of the first octave, and finally for the number of levels per octave of the pyramid. Selected values are: 5 octaves, first octave equal to 0 and 5 levels per octave. Thus, we obtain 250 stable keypoints per image on average.

Since each keypoint descriptor is defined as a 128-dimensional feature vector, there is no algorithm able to identify the exact nearest neighbors in such high dimensional spaces that is any more efficient than exhaustive search. There are some algorithms, such as the Best-Bin-First proposed by Beis and Lowe [5], that return the closest neighbor with high probability. Another useful algorithm is proposed by Arya and Mount [2], based on the use of kd-trees and bd-trees, that support both exact and approximate nearest neighbor searching in spaces of various dimensions. We have used the kd-tree version implemented within the ANN (Approximate Nearest Neighbor) library by Mount and Arya [16] to carry out our experiments.

Our image database contains a set of 45374 data points in real 128-dimensional space. However, with a tree structure, the nearest neighbor can be found efficiently. ANN allows us to select the number of returned k-nearest neighbors, where $k \geq 1$. For $k = 2$, we compute the distance between these two nearest neighbors in order to check whether it is a real match. As Figure 4 shows, we evaluated three different values for the distance ratio $R$ (between the closest and the next closest neighbor), and we reject those matchings with a ratio greater than the specified value. In our scenario, the best results for performance and accuracy are obtained with $R = 0.75$ (close to the $R = 0.8$ suggested by Lowe in [13]).

Another important parameter of our matching algorithm is the number of individual matches required to consider that there is strong evidence of a global image matching. Since the number of features for each image may be high, we




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

**Fig. 4.** Matching parameter analysis. a) Global tree performance; b)Clustering-based trees performance; c) Global tree accuracy; d) Clustering-based trees accuracy

can specify a matching threshold $T$ (a percentage of matches against the number of features per image) to define a match as valid. We have experimented with different $T$ values (x axis of Figure 4), finding that 0.1 is a good value to maximize performance and accuracy. That is, given a query image, we consider that there is match when at least 10% of its features are found in a particular image of the database. It proves that SIFT features are very descriptive.

Once we have chosen the right parameters for an optimal image search, we can show the appropriateness of using a multisensor system to improve the accuracy and the performance of the location estimation process. As we mentioned above, using the clustering technique based on the RSSIs analysis, we divided our scenario into four zones. Figure 4 (a,c), shows the performance and accuracy results, respectively, obtained from a experiment using a tree of images that contains all the images of the database. However, Figure 4 (b,d) shows the performance and accuracy results, but with five different trees. Four of them contain the images of each zone, and the fifth one is a global tree containing all the images in the scenario. Analyzing these results, we can conclude that when using smaller trees we get better performance results, around 25% reduction in the search time, even improving accuracy. We are aware that our scenario is not so large, meaning that in larger scenarios this difference of using a global tree against clustering-based trees will suppose a higher performance improvement and, therefore, a better scalability.

Finally, it is worth noting that SIFT features provide a good distinctiveness even in environments with many similarities between different physical emplacements. Figure 5 shows different images, and their related features, extracted from the experimental environment. Each feature is represented by a circle indicating the corresponding scale and by a vector showing the orientation. Red circles are light blobs on dark background, green ones are dark blobs on light background, and a specific feature is drawn as a blue histogram, for illustration purposes. As Figure 5 shows, despite corridors A and B are similar, there is a significant difference regarding the number of features matching the captured image.

## 6 Motion Analysis

In several situations there is no need to perform continuous estimations about the user location. Once the right position has been determined with a high confidence,

(a) Features (b) Matching with corridor A (c) Matching with corridor B

**Fig. 5.** SIFT features and matching of images

the information obtained by the built-in inertial sensors of a smartphone might be used to determine whether the user moves or remains still. In this latter case, there is no need to acquire additional data from other sensors, or to calculate a new location estimation. The integration of this type of sensors improves the performance of our system, supporting a higher number of users. Additionally, regarding to power consumption, we will be able to save energy and avoid unnecessary operations and transmissions.

We infer the physical state of the mobile phone from a built-in 3-axis accelerometer, so we are able to characterize user movements. Since it is too difficult to get a better motion recognition with mobile phone accelerometers, we decided to identify just two states: still and motion.

To identify these states we use a simple perceptron [6], because we are dealing with two linearly separable sets, and that technique performs reasonably well. In order to train the perceptron, we first got 150 readings from a phone held by a user in a static position. Then we recorded another 150 samples from a user walking with the phone. Once this phase was finished, we trained our perceptron with these 300 samples. We performed this entire process several times using different patterns of movement to obtain different perceptrons. Once we finished this learning stage, we tested the perceptrons in the real scenario and we selected the one providing better results, which was trained with slow movements.

## 7 Sensor Fusion

The sensor fusion of our location system is designed to reduce the computational cost, to minimize the amount of information transmitted and, therefore, to preserve the smartphone battery life. The smartphone uses two different threads running in parallel in order to capture images and to obtain information about RSSI. Access points are analyzed in order to determine whether we are in a location supported by our system. Additionally, we have to check whether the image is useful, that is, it is focused enough to allow the extraction of SIFT features (a gradient model is used to discard blurry and uniform images). In that case we send to a database the image along with the RSSI information and a motion estimation performed locally, otherwise the image is not sent. Finally, once motion is detected or a timer expires, the process is repeated.

Once the information has been stored, the location service is able to estimate the position. RSSIs are processed to get a probability distribution vector indicating the likelihood of being located at each cell of the scenario. SIFT features are computed, and using the ANN kd-tree matching algorithm, a probability distribution is obtained according to the number of matches found.

The matching process proceeds as follows. We select the subtree linked to the cluster containing the cell with higher probability after the RSSI analysis. We compare our input image features with those stored in the selected subtree. If the result does not exceed the matching threshold $T$ (mentioned in Section 5), we select another subtree containing that cell, if available, and the process is repeated. Finally, if we do not exceed $T$ we select the global tree to perform a new search. According to the results shown in Figure 4, the performance of using subtrees in our testbed is up to 25% better than a global tree search.

The fusion process uses Bayes' Rule to obtain a probability distribution indicating the likelihood of being located at each cell of the scenario. Being $\pi$ a probability distribution vector over each cell, $C = \{c_1, .., c_m\}$ the set of cells that make up the finite space state, $n$ the amount of RSSI measurements in the current observation $O_j$, and $Pr(\lambda_\beta|a_\beta, c_i)$ the probability of taking a measurement from the access point $a_\beta$ at reference cell $c_i$ with a signal strength $\lambda_\beta$ , there is a first estimation based only on RSSI:

$$\pi'_i = \frac{\pi_i Pr(O_j|c_i)}{\sum_{\alpha=1}^{m}(\pi_\alpha Pr(O_j|c_\alpha))} \qquad \text{where } Pr(O_j|c_i) = \prod_{\beta=1}^{n} Pr(\lambda_\beta|a_\beta, c_i) \tag{1}$$

Considering $\pi'_h$ as the highest value in $\pi'$, we can constrain the analysis of the image in $Oj$ to the cluster of cells where $c_h$ is included. $Pr(f|c_c)$ is the probability of seeing an image $f$ at cell $c_c$ contained in the selected cluster, and it is defined as follows:

$$Pr(f|c_c) = \frac{matches_{f,c_c}}{\sum_{k=1}^{l}(matches_{f,c_k})} \tag{2}$$

All the images related to that cluster are analyzed to determine the image with a higher number of matching features. If $Pr(f|c_c) > T$ ($T$ was defined in Section 5), we consider that there is strong evidence of being at cell $c_c$. Those cells not related with the selected cluster are assigned a negligible probability value (to avoid zero probability distribution). Finally we recalculate the probability distribution $\pi'$ by fusing the already estimated $\pi'_i$ with the corresponding $Pr(f|c_i)$ probability.

## 8 Experimental Analysis

In order to validate the accuracy and performance of our multisensor system we carried out several realtime tests where users were still and moving. Five different users pertaining to our research group participated in these tests daily during four weeks, making use of a augmented-reality prototype able to display location-aware notes. Every cell in the scenario was linked to a virtual note. Users were able to provide feedback through the application in order to confirm

**Table 1.** Cell hit (including adjacent) using different sensors

| Fused Sensors \ Cell hit | **Still** | **Motion** |
|---|---|---|
| RSSI | 68.75% (93.75%) | 5.5% (38.9%) |
| RSSI + Acc | 56.25% (90.65%) | 5.5% (38.9%) |
| Images + RSSI + Acc | 82.85% (97.14%) | 55% (94.44%) |

their estimated positions, or to provide the right cell when the estimation was not correct, and this was our way to establish the ground truth.

During all these tests we obtained information from all the available sensors (images, RSSI and accelerometer). We have divided the tests into two different categories: still tests where users remain still at the same place, and motion tests where users move along the dependencies. The still tests took place at several cells, in corridors, offices and laboratories. During the motion tests, we covered several paths mainly along the corridors. Location was estimated with three different combinations of sensors: using RSSI only, using RSSI and accelerometer and using all the sensors, in order to check the accuracy of each combination. Table 1 shows that, for motion tests, we are able to estimate the right cell 55% of cases using images (94.44% of the estimations are exact or directly adjacent to the right cell). It is worth noting that the results improve the accuracy obtained using only WiFi, both for still and motion cases.

Regarding to the system performance, Figure 6 shows the distribution of time among the different tasks to be performed during the whole cycle needed to estimate locations. The required mean time is around three seconds using a HTC Desire smartphone and, as mentioned, our main intention was to find a good tradeoff between accuracy and performance. We consider that the extra time required to deal with images is acceptable for some of the envisioned applications requiring better accuracy. Furthermore, most of the time required for some tasks included in Figure 6 will be reduced significantly using future smartphones. For example, RSSI acquisition requires 35% (one second) since we perform a passive scanning. However, devices supporting active scanning might reduce this time drastically. In relation to images, we have considered the required time for auto focus, but this is not necessary once the focus is completed and we want to obtain more pictures. Additionally, communications have an important overload (25%), and most of this time is consumed by the transmission of images. It is worth reminding that we employ accelerometers in order to save energy and calculations in those situations where the handset remains still, and images are discarded when the device is in the pocket or in similar cases. Transmission of images is a requirement due to the limitations of current smartphones to perform the extraction of SIFT features. However, some manufacturers are announcing new smartphones with built-in GPUs, like NVIDIA TEGRA 2, that will be able to perform that extraction locally. This is especially interesting in order to reduce the required time to estimate the position by the location server, which is 13% (around 400ms) but including the calculation of SIFT features.

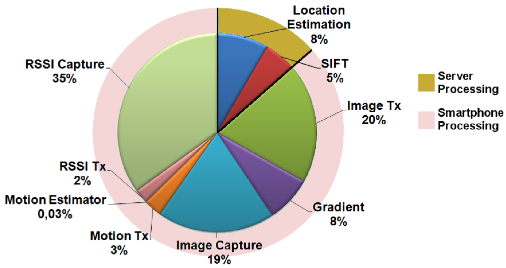



**Fig. 6.** Time latency of the localization process

## 9 Conclusions

The multisensor localization system presented in this paper provides an accurate method for estimating the user location. We have concentrated our efforts on integrating the context information obtained by using the available sensors in current smartphones.

As mentioned during the paper, the main drawback of using images is the elevated computational cost of the matching process in huge scenarios. Though our experimental scenario is relatively small, we have demonstrated that by using RSSI information we can reduce the search time up to 25%. In larger scenarios this performance improvement will be meaningfully higher since the complete tree of features will contain a higher amount of descriptors. Realtime experiments provide good results for still and motion scenarios, respectively, while the system can currently complete an entire cycle, from obtaining data from sensors to providing a location estimation, in approximately three seconds. Additional optimizations have been added by considering the information from the accelerometer and discarding useless images.

There are several augmented-reality applications which do not require a real time response. Therefore, we find our approach a valuable contribution for this kind of location-based services, since we have obtained the required tradeoff between a fine-grained accuracy and an acceptable response time using data from multiple sensors.

**Acknowledgments.** This work was supported by the Spanish MICINN, Plan E funds, under Grant TIN2009-14475-C04-02/01. It was also partly supported by the Funding Program for Research Groups of Excellence (04552/GERM/06) established by Fundación Séneca.

## References

1. Arai, I., Horimi, S., Nishio, N.: Wi-Foto 2: Heterogeneous device controller using Wi-Fi positioning and template matching. In: Proc. Pervasive 2010 (2010)
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In: ACM-SIAM Symposium on Discrete Algorithms, pp. 573–582 (1994)

3. Azizyan, M., Constandache, I., Roy Choudhury, R.: SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In: Proc. of the 15th MOBICOM, pp. 261–272 (2009)
4. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-based User Location and Tracking System. In: Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 775–784 (2000)
5. Beis, J., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Proc. CVPR 1997, pp. 1000–1006 (1997)
6. Block, H.D.: The perceptron: a model for brain functioning. I, pp. 135–150. MIT Press, Cambridge (1988)
7. Chandrasekaran, G., Ergin, M.A., Yang, J., Liu, S., Chen, Y., Gruteser, M., Martin, R.P.: Empirical Evaluation of the Limits on Localization Using Signal Strength. In: Proc. SECON 2009, pp. 333–341 (2009)
8. Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical Robust Localization over Large Scale 802.11 Wireless Networks. In: Proc. MOBICOM 2004, pp. 70–84 (2004)
9. Hattori, K., Kimura, R., Nakajima, N., Fujii, T., Kado, Y., Zhang, B., Hazugawa, T., Takadama, K.: Hybrid indoor location estimation system using image processing and WiFi strength. In: Proc. WNIS 2009, pp. 406–411 (2009)
10. Hile, H., Borriello, G.: Positioning and orientation in indoor environments using camera phones. IEEE Computer Graphics Applications 28, 32–39 (2008)
11. Konolige, K., Chou, K.: Markov Localization using Correlation. In: Proc. IJCAI 1999, pp. 1154–1159 (1999)
12. Lemelson, H., Kjærgaard, M.B., Hansen, R., King, T.: Error Estimation for Indoor 802.11 Location Fingerprinting. In: Choudhury, T., Quigley, A., Strang, T., Suginuma, K. (eds.) LoCA 2009. LNCS, vol. 5561, pp. 138–155. Springer, Heidelberg (2009)
13. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: Proc. ICCV 1999, pp. 1150–1157 (1999)
14. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004)
15. Miyaki, T., Yamasaki, T., Aizawa, K.: Tracking persons using particle filter fusing visual and WiFi localizations for widely distributed camera. In: Proc. ICIP 2007, pp. 225–228 (2007)
16. Mount, D.M., Arya, S.: Ann a library for approximate nearest neighbor searching. In: 2nd CGC Workshop on Computational Geometry (1997)
17. Mulloni, A., Wagner, D., Barakonyi, I., Schmalstieg, D.: Indoor Positioning and Navigation with Camera Phones. IEEE Pervasive Computing 8, 22–31 (2009)
18. Park, J.G., Charrow, B., Curtis, D., Battat, J., Minkov, E., Hicks, J., Teller, S., Ledlie, J.: Growing an organic indoor location system. In: Proc. MobiSys 2010, pp. 271–284 (2010)
19. Sohn, T., Li, K.A., Lee, G., Smith, I., Scott, J., Griswold, W.G.: Place-Its: A Study of Location-Based Reminders on Mobile Phones. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 232–250. Springer, Heidelberg (2005)
20. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: Proc. ICCV 2009 (2009)
21. Vedaldi, A., Soatto, S.: Features for recognition: viewpoint invariance for non-planar scenes. In: Proc. ICCV 2005, pp. 1474-1481 (2005)
22. Werner, M., Kessel, M., Marouane, C.: Indoor Positioning Using Smartphone Camera. In: Indoor Positioning Indoor Navigation (IPIN) (September 2011)