Indoor Localization Using Camera Phones *

Nishkam Ravi, Pravin Shankar, Andrew Frankel,
Ahmed Elgammal and Liviu Iftode
Department of Computer Science, Rutgers University, Piscataway, NJ 08854
{nravi, spravin, afrankel, elgammal, iftode}@cs.rutgers.edu

Abstract

Indoor localization has long been a goal of pervasive computing research. In this paper, we explore the possibility of determining user's location based on the camera images received from a smart phone. In our system, the smart phone is worn by the user as a pendant and images are periodically captured and transmitted over GPRS to a web server. The web server returns the location of the user by comparing the received images with images stored in a database. We tested our system inside the Computer Science department building. Preliminary results show that user's location can be determined correctly with more than 80% probability of success. As opposed to earlier solutions for indoor localization, this approach does not have any infrastructure requirements. The only cost is that of building an image database.

1 Introduction

Most of the early systems for indoor localization [30, 9, 4, 16] focussed mainly on location accuracy and involved the use of custom hardware. This implied heavy deployment costs and labour requirements. Of late, the focus has shifted to minimizing infrastructure requirements without compromising substantially on accuracy [14, 23, 6]. The reason is well understood: since location information only serves as a parameter to location-based services, the cost of deploying localization systems should be a minute fraction of the total cost of provisioning location-based services.

Camera-equipped mobile phones are being put to many uses as an interesting study indicates [12]. In this paper, we explore the possibility of determining user's location indoors based on what the camera-phone "sees". The camera-phone is worn by the user as a pendant(Figure 1), which captures images periodically and sends them to a web server over GPRS. The web server has a database of images with their corresponding location. Upon receiving an image, the web server compares it with stored images, and based on the match, estimates user's location. We accomplish this with off-the-shelf image matching algorithms, by tailoring them for our purpose. We improve location accuracy by using an algorithm that takes into account the trajectory of the user. We built an image database for the Computer Science building with nearly ten pictures per "corner" to account for real-life issues such as varying heights of the users, different angles that may correspond to the same image, etc. Our experimental results indicate that room-level accuracy can be achieved with more than 90% probability, and meter-level accuracy can be achieved with more than 80% probability. The error can be further reduced by using more sophisticated image matching algorithms, which is not the subject of this paper. The key advantage of using this approach is that it does not require any infrastructure. Neither custom hardware, nor wireless access points are required. Physical objects do not have to be "tagged" and users do not have to carry any device apart from what they already do: a mobile phone. The only cost involved is that of building an image database.

2 Approach, Issues and Solutions

2.1 Location Determination

When the web server receives a query image, it compares it with the images stored in the database. Every image that matches with the query image is assigned a weight which reflects the degree of similarity between the two images. By image comparison, we imply feature comparison. We use three off-the-shelf algorithms for image comparison: Color Histograms [24, 28, 27], Wavelet Decomposition [10] and Shape Matching [11]. Each algorithm assigns a weight to the image. The total weight of an image is calculated as a linear combination of the weights assigned by each algorithm. If the

*This work is supported in part by the NSF under the ITR Grant Number ANI-0121416

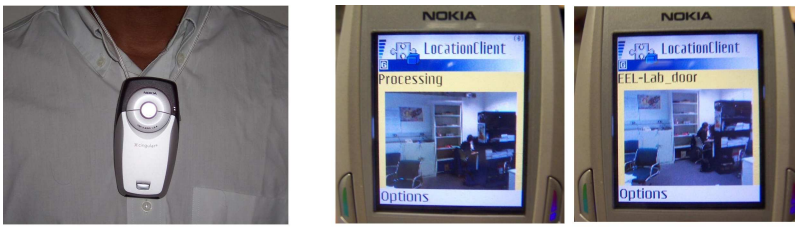

Figure 1. Left: User wearing the phone as a pendant, Right: Snapshots of the client running on the phone

weight of the best match is less than a certain threshold value, the query image is discarded. This is necessary to prevent wrong location updates from being sent to the client. We found that the Color Histograms algorithm worked very well for image comparison as compared to the other two. We assigned a very high weight to the Color Histograms algorithm in our system and low weights to the other two algorithms. Once the weight of the images in the database with respect to the query image are known, the following methods can be used for location determination:

Naive Approach: In this approach, the images in the database are organized in a flat manner. The location of the user is the location of the image that matches the query image with maximum weight. In other words, the location of the user is the one that maximizes the probability of seeing the query image.

Hierarchical Approach: In this approach, the images in the database are organized hierarchically. The images corresponding to a floor are grouped together, the images corresponding to a room are grouped together and so on. When the system determines that the user has entered a particular room, subsequent searches are performed only on the images of that room, until the system discovers that the user has exited the room. The advantage of using this approach is that the probability of error decreases, because the system has fewer images to confuse the query image with. The disadvantage of this approach is that if the system incorrectly determines the room the user is in, subsequent searches would get affected and produce wrong results.

History-based Approach: In this approach, the web server keeps track of the trajectory of the user. Based on the past locations of the user, a better estimate of the current location can be derived. In other words, the location of the user is determined not from a single query image, but multiple query images received over a certain period of time. When the server receives a query image, it looks at the last *n-1* query images. The current location of the user is the one that maximizes the probability of *seeing* the *n* query images in the shortest period of time.

To implement this, we use a simple shortest-path/nearest-neighbors approach. Upon receiving the *nth* query image, the web server carries out a search (image comparison) based on the last *n* query images. Let k_i denote the set of top ten images that match with the *ith* query image, and let $location(k_i^j)$ denote the location of the *jth* image in that set. The weighted euclidean distance between the *jth* image of set k_i and the *mth* image of set k_{i+1} is given by :

$$d_i^{j,m} = w * |location(k_i^j) - location(k_{i+1}^m)|.$$

Where w is inversely proportional to the weight of image k_i^j.

$Path$ is the set of images (one corresponding to each query image) such that the sum of the weighted euclidean distances between these images is the minimum among all possible combinations. In other words, these images define the shortest possible path (scaled by the weight of images) that the user could have traversed, and also the most likely.

$$Path = \{\forall_{i=1}^{n} k_i^j | D_i^{j,k} = d_i^{j,k} + D_{i+1}^*, D_i^* = min_{j,k} D_i^{j,k}\}$$

The location of the image in set k_n that belongs to the set $Path$ is returned as the current location of the user:

$$current_location = location(k_n \cap Path).$$

We use the *sliding-window* approach. The window of *n* images to be considered for calculating the shortest-path slides by one with every new query image. The intuition behind using the shortest-path/nearest-neighbors approach is the following: assuming that the user does not abruptly increase her walking/running speed above a certain limit, the current location of the user should

Figure 2. Two low-resolution images with different locations corresponding to the same corner. First image was taken with camera in the south-east quadrant of the room. Second image was taken with camera in the north-east quadrant of the room.

not be too far away from her past locations. We use weighted distance in order to give higher priority to images with higher weight (i.e images which match better with the query image). This is the reason for using a multiplicand *w*, which is inversely proportional to the weight of the image.

2.2 Database Creation

For good results, it is important to build an extensive database of images. We wrote a Java client (createDB) that runs on the phone and sends images as the database creator walks around at very low speeds, so as to allow multiple images of a *corner* to be taken. The web server extracts features from the received images on the fly and stores the features in a database. The image may or may not be stored. The images/features in the database are tagged with location, manually afterwards. The process of tagging images with location can be partially automated, by using a speech recognition interface on the phone, so that the database creator can tag images by announcing her location while pictures are taken.

During image database creation, we take multiple images of the same *corner* to accommodate for different heights and angles. On an average we store ten images per corner. The success of the system depends on the amount of effort put into database creation. More images per corner increases the chance of success. The number of images in the database is also determined by the desired frequency of location updates. High update frequency requires high coverage which implies a dense image database to accommodate for small changes in the location of the user.

2.3 Energy Optimization

Energy consumption on the phone is primarily determined by two factors: frequency of sending query images and size of the image. We found that as long as the resolution of images in the database is the same as the resolution of query images, performance is unaffected. We worked with low-resolution images of size 5KB and could save substantially on energy consumption. For reducing the frequency of sending query images without affecting the frequency of receiving location updates, we came up with a *server-initiated* location query approach. In this approach, when the server responds with location, it piggybacks the time period after which the phone should send the next query image. The value of the time period is determined by the current speed of the user and his distance from the next *critical point*. Critical points are those where the user can change directions (e.g intersections of corridors). The server continuously sends back location updates by *guessing* the location of the user based on their speed and last location. For details refer to [19]. Note that this approach is applicable to only corridors and such, and is based on the assumption that the user does not abruptly change his speed and direction.

Table 3. Robustness of the three approaches (number of erroneous images that result in incorrect location)

Approaches	Average
History-based	2.67
Naive	1
Hierarchical	1

3 Experiments and Preliminary Results

Client-side components were implemented in Java using MMAPI [1] which is supported on some Symbian OS phones. We used the Nokia 6620 mobile phone. All server side components were implemented in C++ for performance reasons. We created a partial image database for the third floor of the Computer Science Department building, to include sixteen rooms, staircase,

Table 1. Probability of success for the three experiments

Approaches	Naive	Hierarchical	History-Based
Room-level accuracy	93%	N/A	N/A
Quarter-room-level accuracy	83%	96%	94%
Corner-level accuracy	50%	Non-deterministic	80%

Table 2. Energy consumption and response time

Image Size	Avg. Response Time	Avg. Energy Consumption
5KB Image	720 msec	630mJ
128KB Image	4100 msec	3600mJ

a bridge and the corridors. Creating the database is a slightly tedious process. It takes around an hour to construct the database for a room which includes the time for database construction, pruning as well as annotation of images. We constructed the database over multiple days. Database construction as well as experiments were carried out during the day on weekends. In all, we have around 300 locations (i.e corners) covered in the database. The goal of the experiments was to test the feasibility of our approach. We therefore sought the answers to the following two questions : (1) How successful is our approach in achieving room-level accuracy?, and (2) How successful is our approach in estimating the orientation and location of the user anywhere in the building?

We conducted three main experiments. The experiments were carried out on two users with five trials per user. In the first experiment, the user would wear the phone as a pendant and enter the room as the camera took a picture. The image database would comprise of only pictures corresponding to user standing at the door of a room and facing inside, with ten images per door. This experiment was conducted to find the probability of success for room-level accuracy. Hierarchical and History-based approach does not apply to this scenario. Figure 3 shows the pictures of a few rooms that we experimented with. Figure 4 shows an example query image that matches best with Image 3 in Figure 3. In the second experiment, the image database comprised of only inside-room pictures of all rooms. The experiment was conducted with the user wearing the phone as a pendant and walking around inside rooms, sending a query image every four seconds.

Figure 5 shows inside-room pictures of a few rooms. Figure 6 shows example of a query image that matches best with Image 4 in Figure 5. Most of the rooms in the Computer Science building are around 4mx7m. We report user's location to quarter-room level accuracy (North-East, North-West, South-East, South-West), and report one of the four orientations (facing North, facing South, facing East, facing West). The database consists of ten images per corner to account for varying user heights and angles, and sixteen corners per room (four corners per quarter corresponding to the four orientations). The third experiment was conducted with the complete database, as the user walked around on the 3rd floor. This includes corridors, staircase and the bridge in addition to rooms. We call this *corner-level* accuracy. During the experiments, the users would either stand stationary when the image was being clicked or walk at very low speeds, in order to cover all the corners. This was necessary because of GPRS latency.

Table 1 shows the results for the three experiments. The results correspond to the ten trials. From the results, it can be concluded that for applications that require only room-level accuracy, the Naive approach would suffice. Corner-level accuracy corresponds to the most general case: the user walking inside rooms, across rooms and inside corridors. History-based approach clearly shows improved performance for this case. Hierarchical approach performs well most of the time but completely fails sometimes and is therefore not reliable.

We found that the Color Histograms algorithm worked very well as compared to the other two, and was chosen as the primary method of image comparison in our system. We noticed that while most of the images resulted in accurate location, some of them were hard to recognize and were often confused with other similar images, bringing down the overall success rate. We noticed that even when a query image was incorrectly matched, the correct one was often among the top three matches, suggesting that if our approach was combined with another low-cost localization mechanism, extremely high accuracies could be achieved.

We worked with low-resolution JPEG images of size 5 KB. Our experiments with high-resolution images (128 KB in size) resulted in approximately the same success rate as low-resolution images. By working with

Figure 3. Low-resolution pictures of a few rooms taken from the door

Figure 4. Low-resolution query image matches with Image 3

low-resolution images we could save substantially on energy as well as response time. Table 2 shows the average response time (i.e latency) for our experiments, as well as the average energy consumption on the phone for sending a query image and receiving one location update. In the idle mode, the phone consumes around 4mJ per second.

We also measured the robustness of the three approaches. Robustness is measured by counting the number of erroneous images it takes to result in an incorrect location. A higher number implies better robustness. Robustness is a useful measure to account for anomalies, such as user's hand covering the camera lens, or the camera getting deflected towards the roof or the floor. Table 3 shows the value of robustness for nine trials. The average value of robustness for the History-based approach is 2.67. For Naive and Hierarchical approaches, robustness would be 1.

We carried out an experiment to study the effect of changes in the environment, namely: (1) movement of furniture, (2) presence of human beings and (3) varying lighting conditions. This experiment was conducted for one room. We found that slight movement of furniture, such as chairs, computers, objects on tables and shelves did not affect the accuracy of results. The accuracy of results suffered when there was a significant change, such as removal of a desktop monitor from a desk. The accuracy fell drastically when lighting conditions were varied. When the lights in the room were dimmed, the success probability fell to less than 20%. Much to our surprise, the presence of human beings in the pictures did not affect the accuracy significantly. Out of the sixteen corners in the room, only six suffered from the presence of a human being. Three of these were recognized with a success rate of only 10%, two were recognized with a success rate of 50%, and one was recognized with 90% success rate. For the last one, the success rate fell to 70% when the person in the picture wore a brown jacket. (Figure 7 shows the three pictures: the original picture, picture with a person in it, picture with a person wearing a brown jacket.)

We do not completely understand why some images are recognized more accurately than others in the presence of a human being, but we conjecture that it is because of the fact that some images have more distinguishing objects in them as compared to others. We noticed that when an image is recognized with low accuracy due to presence of a person, it is almost always confused with an image of the same corner but taken from a distance. Some of these observations can be attributed to our primary method of image comparison which uses Color Histograms. Color histogram of an image is produced by discretising the colors in the image into a number of bins, and counting the number of image pixels in each bin. Color histogram of an image varies only slightly with the angle of view and the movement of objects inside the image.

4 Related Work

A number of indoor positioning systems have been built. Most of these require special hardware such as ultrasound transmitters/receivers [9, 16], IR badges [30], microphones [23], PCs [6] or use wireless access points to determine location [4, 14]. Microsoft's EasyLiving [13] project uses cameras installed in rooms to track humans using vision techniques.

Work is being done in using camera phones as interaction devices by tagging physical objects with visual codes and using vision techniques to extract and inter-

Figure 5. Low-resolution pictures of different corners

Figure 6. Query image matches with Image 4

pret the information stored in these visual codes [20, 22, 26]. Localization could also be possibly achieved with this method. However, physical objects would have to be tagged. Sarvas et al [21] had demonstrated the use of camera phones for getting meta-information about physical objects using a human-in-the-loop approach. They demonstrated their approach for getting meta-information about outdoor landmarks. We demonstrate the feasibility of using camera phones for indoor localization and without user involvement.

Some augmented reality systems employ vision techniques for augmenting physical objects with meta-information [2, 29]. The goals as well as the approach are quite different. We use image matching algorithms, while augmented reality(AR) community uses object recognition algorithms for getting information about objects in sight. Determining similarity between two images is easier than recognizing objects inside an image.

A large body of work on vision-based robot localization exists in the artificial intelligence(AI) community [25, 8]. However, there are certain key differences. First, most of the work in the AI community has focussed on the use of landmarks for positioning. The robot determines its position based on the coordinates of the landmarks that are visible. Our approach does not make use of landmarks for localization. Second, robot takes an action based on where it is (e.g RoboCup soccer contest [3]). Therefore, centimer-scale accuracy is desired. Besides, the algorithm needs to be computationally very efficient to minimize the reaction time of the robot. Most of the research in the AI community is driven by these issues. Third, most of the work in robot localization is of theoretical nature; experimentation is limited and is done with robots that are very small in size (usually less than a feet tall). These robots do not see the environment the way a human being can by virtue of being tall.

5 Conclusions and Future Work

In this paper we showed that it is feasible to achieve indoor localization using just camera phones and off-the-shelf image comparison algorithms. We were able to attain room-level accuracy with more than 90% chance of success and corner-level accuracy with more than 80% chance of success, using a history-based location determination approach. The latency of receiving location updates over a GPRS connection is around a second, which would be even lower for a 3G connection. We discovered that the image comparison algorithms remain unaffected if resolution of the images in the database is the same as the resolution of the query image. By using low resolution images, we were able to reduce energy consumption and response time significantly.

There are several questions to be answered: How well can this approach scale across buildings, especially for ones with high symmetry? Can this approach be made completely resilient to changes in the environment, such as varying lighting conditions, presence of moving objects such as human beings, movement of furniture etc? Will it be necessary/feasible to combine this approach with other low-cost location sensing mechanisms (such as RFID) to improve accuracy and scalability? More experimentation and better image comparison algorithms are required to answer one or more of these questions. Prior work shows that activity, direction as well as the the speed of the user can be estimated from accelerometer data [7, 5, 15, 18, 17]. Using a body-worn accelerometer in addition to a camera phone may improve location accuracy. The biggest challenge is to minimize the complexity of generating and maintaining the image database. Equally challenging is optimization of energy

Figure 7. Left: image in the database; Center: image with a person (success rate= 90%); Right: image with a person wearing a brown jacket (success rate=70%)

consumption on the phone.

References

[1] Mobile Media API(MMAPI), http://java.sun.com/products/mmapi/index.jsp.

[2] Augmented Reality, http://www.augmented-reality.org/.

[3] RoboCup Soccer, http://www.robocup2005.org/home/default.aspx.

[4] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, 2000.

[5] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceeedings of the 2nd International Conference on Pervasive Computing*, pages 1–17, 2004.

[6] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp. Walrus: wireless acoustic location with room-level resolution using ultrasound. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005.

[7] F. Foerster, M. Smeja, and J. Fahrenberg. Detection of posture and motion by accelerometry: a validation in ambulatory monitoring. *Computers in Human Behavior*, pages 571–583, 1999.

[8] J. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods, 1998.

[9] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking*, 1999.

[10] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995.

[11] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database. In *In Proceedings of International Conference on Pattern Recognition*, 1992.

[12] T. Kindberg, M. Spasojevic, R. Fleck, and A. Sellen. The ubiquitous camera: An in-depth study of camera phone use. *IEEE Pervasive Computing*, 4(2), April-June 2005.

[13] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, 2000.

[14] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of the Third International Conference on Pervasive Computing*, 2005.

[15] S. Lee and K.Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, pages 24–32, 2002.

[16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000.

[17] C. Randell and H. Muller. Context awareness by analysing accelerometer data. In B. MacIntyre and B. Iannucci, editors, *The Fourth International Symposium on Wearable Computers*, pages 175–176. IEEE Computer Society, 2000.

[18] N. Ravi, N. Dandekar, P. Mysore, and M. Littman. Activity recognition from accelerometer data. In *Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI)*, 2005.

[19] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode. Indoor localization using camera phones. Technical report, Department of Computer Science, Rutgers University, October 2005.

[20] M. Rohs and P. Zweifel. A conceptual framework for camera phone-based interaction techniques. In *Proceedings of the Third International Conference on Pervasive Computing*, 2005.

[21] R. Sarvas, E. Herrarte, A. Wilhelm, and M. Davis. Metadata creation system for mobile images. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004.

[22] D. Scott, R. Sharp, A. Madhavapeddy, and E. Upton. Using visual tags to bypass bluetooth device discovery. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1), 2005.

[23] J. Scott and B. Dragovic. Audio location: Accurate low-cost location sensing. In *Proceedings of the Third International Conference on Pervasive Computing*, 2005.

[24] M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1), 1991.

[25] S. Thrun, D. Fox, W. Burgard, and F. Dallaert. Robust monte carlo localization for mobile robots. *Artif. Intell.*, 128(1-2), 2001.

[26] E. Toye, R. Sharp, A. Madhavapeddy, and D. Scott. Using smart phones to access site-specific services. *IEEE Pervasive Computing*, 4(2), 2005.

[27] A. Vellaikal and C. Kuo. Content-based retrieval using multiresolution histogram representation. *Digital Image Storage Archiving Systems*, 2602, 1995.

[28] R. B. W. Niblack and et al. The qbic project: querying images by content using color, texture and shape. In *In Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, 1993.

[29] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmente reality on handheld devices. In *Proceedings of the Third International Conference on Pervasive Computing*, 2005.

[30] R. Want, A. Hopper, V. Falco, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10, 1992.