Positioning and Orientation in Indoor Environments Using Camera Phones

Harlan Hile Gaetano Borriello
Dept. of Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350 USA
harlan@cs.washington.edu gaetano@cs.washington.edu

November 16, 2007

Abstract

Increasingly, cell phones are used to browse for information. Location technologies provide a way to focus on gathering information that is appropriate to the user's current location. We seek to extend this by overlaying information directly onto the physical environment using the cell phone's camera. We process an image from the camera to obtain accurate location and orientation information starting from a rough location estimate obtained from radio signals (e.g., Wi-Fi). The camera's pose can be determined by matching detected landmarks to building database, enabling information to be aligned and overlaid directly onto the camera image and displayed on the phone. We present our system for use in hallway environments and analyze aspects of its performance. We have achieved the accuracy required by our augmented reality approach and computational latency is already within a range suitable for interactive use even though several potential optimizations are still left to be explored.

1 Introduction

As someone walks through a building, he may want help navigating to the right room or accessing dynamic directory information related to his current location. Both of these tasks would benefit from the environment directly providing personalized information. Current approaches to this problem use location systems to help the user index into the information. However, the retrieved information must be put into context. We use a camera phone to provide the contextual framework by having the act of pointing the camera form the query into the information. The phone's display can then be used to overlay information directly onto the image and provide the user information in context. The crucial element of this approach is in determining the precise camera pose (location in 3D and orientation - a full 6 degrees of freedom).

We are working on two applications of this capability. The first is motivated by an ongoing project to help individuals with cognitive impairments navigate in indoor spaces. This user population often has difficulty navigating complex buildings such as medical centers and shopping malls. In essence, we are trying to provide customized "painted lines on the floor" for users to follow to their destination. Our goal is to overlay directional arrows and navigation instructions onto the image as it is easier to understand directions when they are overlaid directly on the user's own view of his environment, especially for people with cognitive impairments [1]. The inability to get around efficiently can limit integration into the community or affect their ability to be gainfully employed. We are building a system that supports both indoor and outdoor navigation; here we focus only on the indoor portion. The second application targets a more general population that may be interested in finding out information about a building such as what events are taking place, which resources are reserved and by whom, and when someone was last in their office. This is what we mean by "dynamic directory information". As people walk down hallways, they should be able to see customized "dynamic name plates" that provide this data. Examples of information overlay for both of these uses are shown in Figure 1.

Our approach to finding the camera pose is based on a simple concept: we determine the "landmarks" in the image and their correspondence to previously cached landmarks of the space. By matching enough landmarks we can precisely compute the camera pose and thereby accurately overlay information onto the display. This simple idea is complicated by the fact that different spaces have different types of landmarks. The landmarks that are available in hallways are generally not available in open areas, such as large rooms. Our previous work introduces our approach to both these problems [2]. In this article we will discuss only our hallway system in further detail.

In hallways, corners, floor-to-wall transitions, and doors are likely to be clearly visible, however there is a high degree of homogeneity between the features. This means an individual feature cannot be uniquely identified, however it is possible to use the pattern of features in an image to determine location by comparing them to known feature locations. Our image processing locates these micro-landmarks in an image and compares them to a floorplan provided by the building's infrastructure. We use a building server to hold this floorplan data as well as provide the computation cycles for extracting the micro-landmarks and performing the matching. Communication between the client and server is realized through a Wi-Fi connection supported by many newer phone models. We prune the search of the best correspondence by using the Wi-Fi fingerprint to coarsely locate the user (we only expect a location estimate that is accurate to within 5-10 meters - easily attainable with several of today's Wi-Fi-based positioning systems [3]). In outdoor environments, we could use GPS-based positioning available on many newer cell phone models.

Thus, our system works as follows: (0) the phone captures the image, (1) sends it to the server along with Wi-Fi fingerprints and the type of information requested, (2) the server performs feature extraction and (3) finds the correspondence to the building's floorplan or to previously captured images, (4) from this correspondence the camera pose can be computed, and finally, (5) the server returns an information overlay for display on the phone client. Figure 2 provides a diagram of how our system operates. Our challenge is

Figure 1: Sample of possible information overlay on an image of the environment. Image shows both an overlaid navigation aid and a magic lens-type application with dynamic information about the current surroundings.

to ensure that all this computation and communication can be performed fast enough to support reasonable user interaction speeds. Although we currently do all the processing on the building's server our goal is to explore different partitions of tasks and evaluate their performance. Moreover, we can further explore the types of features/landmarks we extract from the image and consider several hybrid approaches. We also plan to investigate the use of video (as the user moves the phone) as this may provide more cues than a static image, and be better suited for this type of "magic lens" or augmented reality application. We expect that our current single image technique can be extended to support video.

In the remainder of this article, we examine the steps of processing an image and matching it to a floorplan, illustrated by examples, and analyze the performance of our current algorithms. We will also discuss work on similar problems and how they related to our system. Lastly, we discuss remaining issues and extensions to our system.

2 System

As previewed in Figure 2, the steps of the system will be first to extract possible relevant features from a given 640x480 cell phone camera image. Next a relevant section of the floorplan (or other feature database) must be chosen, and then the matching is performed. Once the correspondence is found, it is used to find location and orientation of the camera. Lastly, the location of objects and other relevant information is translated back into image space. The following sections will discuss these steps in more detail, illustrated by examples.

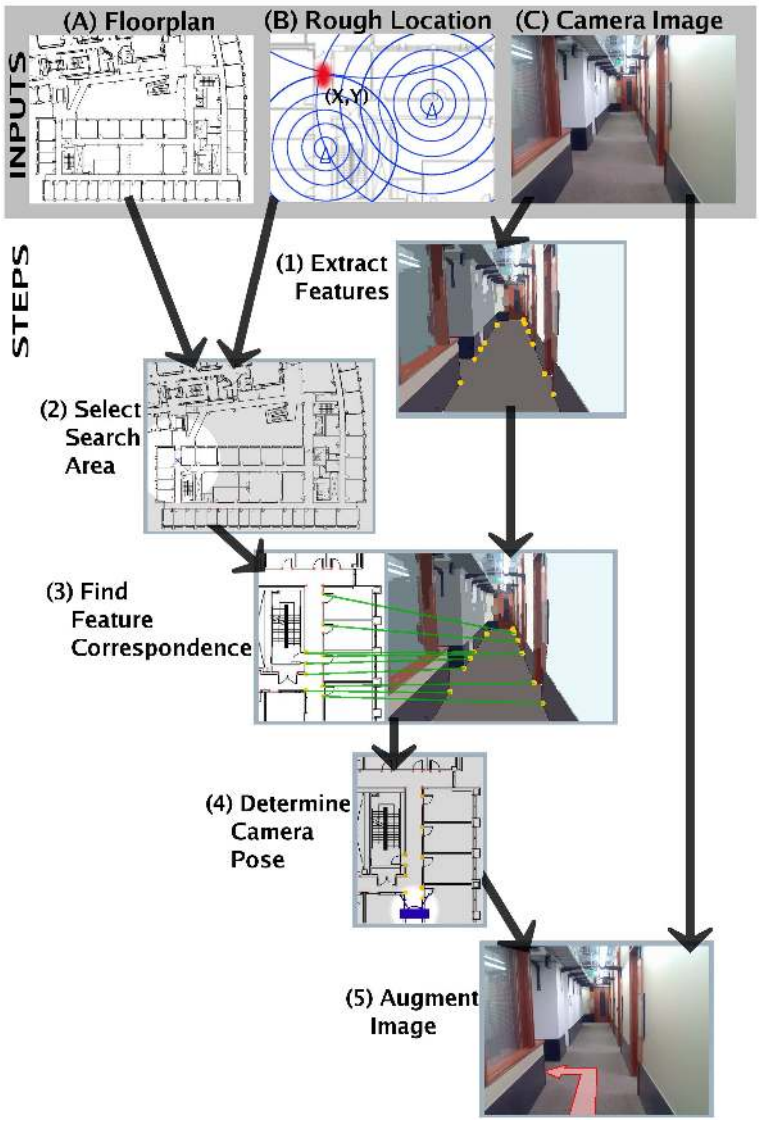

Figure 2: System diagram for calculating camera pose and overlaying information on an image from a camera phone. The inputs to the system are the following: (A) a floorplan with relevant features marked on it, (B) a rough location estimate (which includes floor information), and (C) an image from the camera. Using this information, the first step is to extract relevant features from the image. Step two is to choose a region of the floorplan and a set of features to match against. In step three, a mapping between the features in the image and features on the floorplan must be assigned and evaluated. In step four, this information can then be used to find a more precise location and orientation on the floorplan. Lastly in step five, location of objects and features on the floorplan can then be transferred onto the image, along with data relevant to the user. A similar methodology exists for using textural landmarks in open areas. [2]

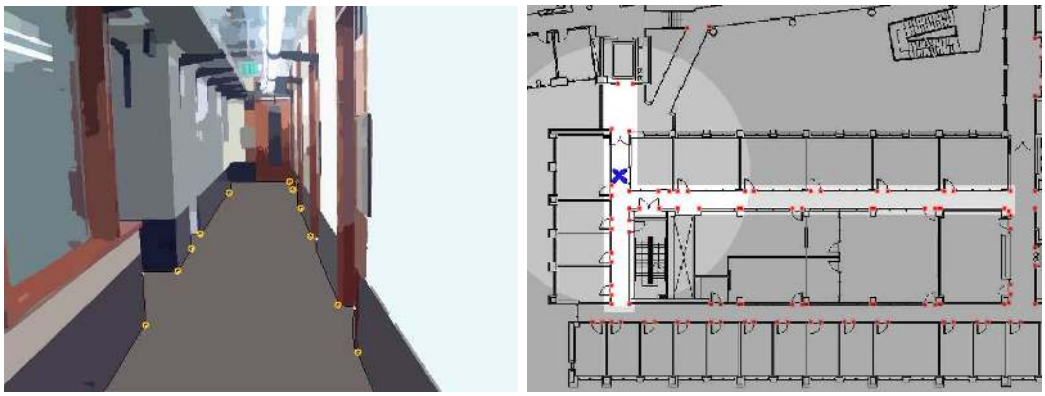

Figure 3: Left: The hallway image is first segmented with Mean Shift, then the edge of the floor traced and corners located. Corners marked with orange circles are candidates for matching to floorplan features. Right: The region of the floorplan considered for matching. The estimated location is marked with an X, and the feature points are marked with small red dots. The correspondence between these two sets of features must be found.

2.1 Feature Detection in Hallways

The first step in finding how an image matches to a floorplan is to locate the features in the image. There is limited information on a standard floorplan, but any floorplan should include location of doorways and corners of walls. These features, or micro-landmarks, will be visible in the image too, and the goal is to find them. The concept is simple: to locate the lines that define the edge of the floor and the lines that define the edge of each doorway or corner. Intersecting these sets of lines will give points that correspond to the features on the floorplan.

We have implemented a basic feature detection method based on segmenting the image. Instead of looking for the edges of the floor directly, we locate the entire floor and then use the edges of that region. We have chosen the Mean Shift method to perform image segmentation [4]. The floor will then likely be the segment that is at the bottom center of the image. It is difficult to take a picture of a hallway where this is not the case, but this requirement could be included in the interface that prompts users to take pictures. The edge of this region is then traced and the corners are identified using a "cornerity" metric. This will not locate all the places along the floor where there is a doorway, but it finds many of them. More can be located by intersecting vertical lines found in the image with the floor boundary, but this is not done in these examples. Additionally, this tracing method finds some false corners that do not correspond to anything in the floorplan. False corners that are at the top of a vertical line of the floor edge can be discarded as points likely caused by occlusion of the floor. Although this basic feature detection gives results suitable for demonstrating the system, we are looking at other methods to improve both speed and robustness. The results of segmentation and corner finding are shown in the left of Figure 3. These are points that can now be used to match to the floorplan.

2.2 Feature Matching

Once the features in the image are found, they must be matched to the floorplan. The first step of this is to choose a set of points from the floorplan to match against. This is done both to remove ambiguous cases and to reduce the search space. The rough location estimate provides a center for the region to be tested, and a radius can be estimated based on the accuracy of the location system and some idea of the

camera's useful visibility range. Figure 3 shows the floorplan on the right, an estimate of the location, and the features that will be considered for matching for the hallway image shown on the left of Figure 3. Since the number of points to consider influences the speed of the algorithm, instead of including all points within a radius, we approximate a visibility calculation by only including points of hallways that are nearby the camera. It is possible to include other information when calculating the region to consider, such as true visibility calculations, or priors for direction, either from direction of motion or an external sensor, but we currently do not include these.

Once the two sets of features are defined, a correspondence must be determined. This is a challenging problem because of the number of possible ways to match. A transformation between the image space and the map space can be defined by four sets of correspondences. For the first example presented here, there are 10 image points and 32 map points, resulting in over 4 billion possible four-point correspondences. In order to make this problem tractable, we perform this matching using a RANSAC (RANdom SAmple Consensus) approach that intelligently selects and prioritizes the hypotheses. The hypotheses are generated using minimal structural assumptions. Lines are fit to the features in the image space, which should produce one line corresponding to the left side of the hallway and one line corresponding to the right side. Two lines containing features are randomly chosen from the floorplan (for most cases there are only two lines), and a direction (relative to the lines and estimated camera position) is also randomly chosen. Now two points are randomly chosen from each line in such a way that ordering of the points along the lines is consistent. Hypotheses are also prioritized based on the area covered by the four points in the image. The larger the distance between the points, the more likely it will produce a stable homography. The area of the bounding box of the four points is used as a measure of spread, and a threshold is slowly lowered as a prioritization method. Lastly, the search is terminated early if the estimate has not improved in a "long time" (for example, 5,000 samples) because it unlikely to improve further as the threshold lowers. Our examples obtain good results testing less than 10,000 hypotheses.

Hypotheses are evaluated by solving for a camera pose that maps from the image points to the floorplan points, and looking at the sum of the squared distance between the two point sets. Weights for points closer to the camera (lower in the image) are higher because they are likely to be detected more accurately in the image. Additionally, unlikely camera parameters (such as a skewed image plane, or large distance from the estimated location) can be penalized to improve the solution ranking. The highest ranking solution at the end of the search is then used. See Figure 4 for results from our Computer Science building. Figure 5 shows results from our Health Sciences Center. This image presented more challenges due to the lower contrast, reduced brightness, and reflective floors. Despite these complications, enough features were detected to produce an accurate match. Improvements in feature detection are still necessary to deal with this type of challenging environment more robustly.

The RANSAC method described here returns results in 3 to 6 seconds on a 2.8GHz desktop machine, and is the slowest part of our current pipeline. We believe that further improvements are possible that will increase both speed and robustness, however our system already approaches speeds near our goal to support interactivity. Performance of our current system will be discussed further in Section 3 and possible extensions will be discussed more in Section 5.

2.3 Augmenting Images

During the process of finding the correspondence between the image features and the floorplan landmarks, both the camera location and orientation are solved for simultaneously. The increased location accuracy and camera orientation information can be leveraged by applications when determining what information to display. The mapping between the image and floorplan can then be used to overlay information onto the camera image. Arrows to give navigation directions can be drawn on the floor by drawing the arrow on the

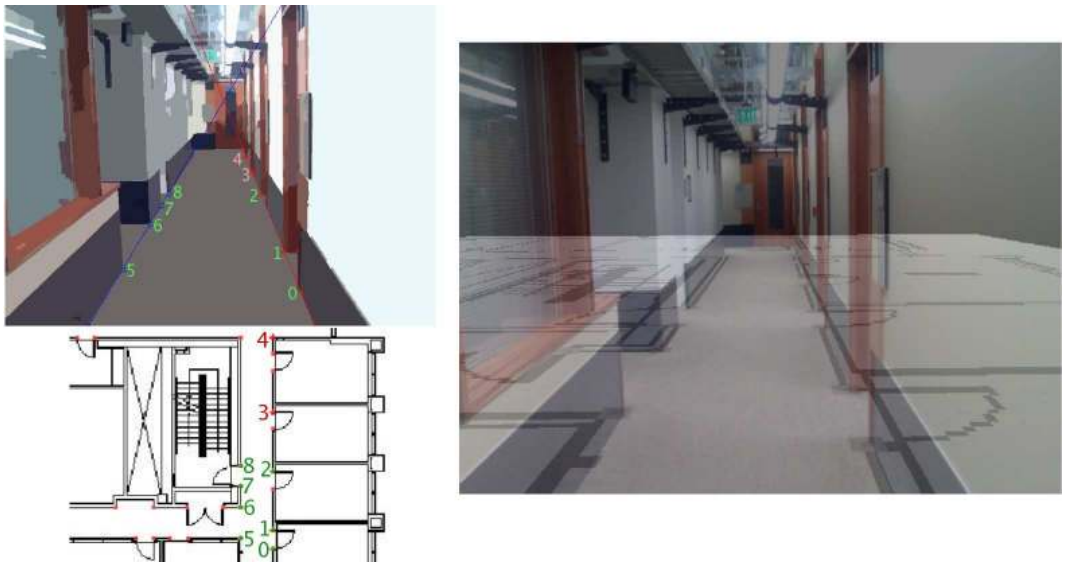

Figure 4: Results of point correspondence algorithm. Upper left: the detected points on the image labeled by number. Lower left: the points on the floorplan that were matched to the image, labeled with corresponding numbers. Right: the floorplan warped into the image space and overlaid on top of the original image, which matches very well. This example has 10 image points and 32 map points, and completes matching in about 4 seconds.

Figure 5: Results of point correspondence algorithm. Upper left: the detected points on the image labeled by number. The detected points of this example required some manual filtering to get a good match. Lower left: the points on the floorplan that were matched to the image, labeled with corresponding numbers. Right: floorplan warped into the image space and overlaid on top of the original image. This example has 9 image points and 19 map points and completes matching in about 2 seconds.

Figure 6: This shows an arrow overlaid to match the perspective of the floor. The calculated homography was used to place the arrow.

floorplan and warping it into the image space. Tips of arrows can even disappear behind corners to give an added sense of depth by clipping to the area segmented as floor, without requiring additional knowledge of the 3D structure. An example of this is shown in Figure 6. Since the location of doors is known in the image, the doors can be marked with additional labels, as suggested in Figure 1.

3 Analysis

In this section we present some analysis of our current system. Although our system is not yet mature, we believe this analysis demonstrates its usefulness and provides motivation and direction for further improvements. Our system can currently complete an entire cycle, from taking a picture on the phone to displaying an augmented image on the phone, in approximately 10 seconds, using a standard 2.8GHz desktop computer to do the processing. About one second of this time is spent on image transfer (over 802.11). The remainder is spent on the processing done by our system. Our current feature extraction first performs Mean Shift segmentation (1.5 seconds) followed by edge and corner location (1 second). Then the features correspondence is determined (3-6 seconds), and the image is augmented (0.5 seconds) before sending it back to the phone for display. In the remainder of this section we investigate different aspects of this performance. We mainly focus on the correspondence matching since it requires the majority of the processing.

3.1 Speed and Accuracy

In order to evaluate the speed and accuracy of our system, we measured the localization accuracy as the RANSAC algorithm was running, and averaged these numbers over 200 trials. The results from our current implementation are shown in figure 7. They show that there is variation among examples in different locations, but on average, errors drop to under 30 cm in only a few seconds, with little improvement afterwards. Although the stopping condition in our application does not use a strict time cutoff, this shows that on average a highly accurate solution is reached quite quickly. This 30 cm accuracy is far greater than what is available from the current Wi-Fi or GPS based positioning systems. Additionally, we have found that even when the distance error is greater, the alignment between floorplan and image that is calculated is often visually acceptable for producing information overlays. While these results are promising, we would like to explore other matching systems that may allow for easier inclusion of constraints or prior knowledge, or may

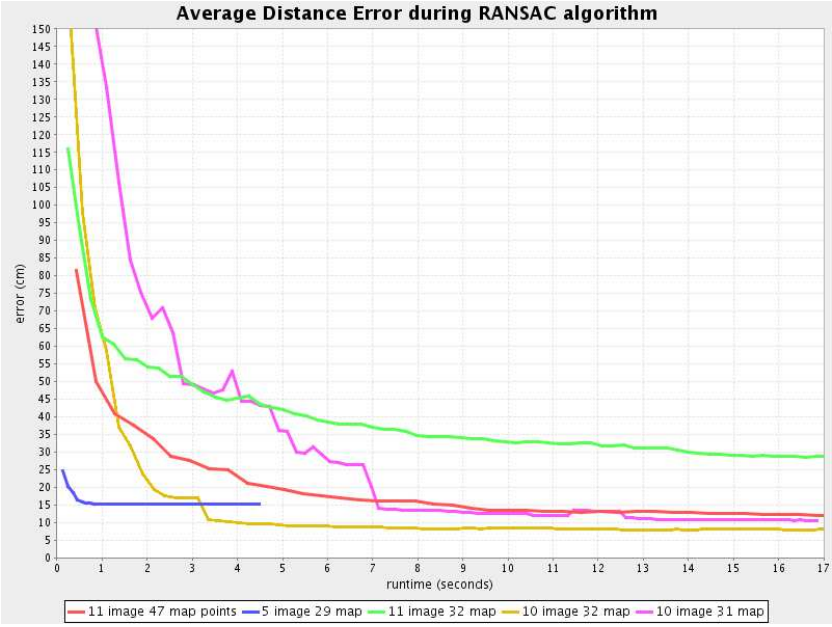

Figure 7: An analysis of average localization error as the RANSAC algorithm progresses. This compares examples from several areas of a building with different numbers of points detected in the image and different numbers of points in the map region of interest. As seen here, location error drops quickly and is under 30 cm in only a few seconds.

have a better guarantee on convergence.

3.2 Occlusion of Features

In many environments, there may be clutter or lighting conditions that make it impossible to detect all the micro-landmarks in a region. It is desirable that a solution degrade gracefully as the amount of data decreases. As shown in figure 8, this is the case for our method. Even with almost half of the possible points excluded, it can still get reasonable results. This means that if the feature detector cannot detect all points because of occlusions or other difficulties, localization can still perform quite well. However, this does depend on which points are missing; for example, if only points from one side of the hallway are detected, our system cannot produce a result from a set of co-linear points. This can be seen in the histogram in Figure 8 as the number of trials with no solution increases as more points are occluded. However, in our example images, although the feature detector does not locate all possible micro-landmarks, the distribution is sufficient to allow localization. It may be possible to achieve reasonable results with fewer points detected if we also make use of line constraints or can constrain the camera pose by other means (either from other sensors, by image analysis, or by assumption).

In addition to missing features, our feature detection algorithm may also detect false features. For this reason, our RANSAC algorithm does not require all the detected features to be matched to floorplan features. Although we have not produced quantitative measures, we have found from various examples that these distracter points are handled well as long as approximately 80% of the points detected are true features. A more intelligent feature detector could also help this situation by classifying the quality of detected features.

Figure 8: Left: an analysis of error and speed as points are randomly dropped from 200 trials of an example, simulating occlusion. Right: a histogram showing distribution of errors for each of the examples at 5 seconds of runtime. Note that our current feature detector found 10 of the 14 features for this example. Average error gradually increases as points are dropped from the example, and the likelihood of no solution increases as points are dropped. This shows a graceful degradation producing reasonable results with nearly half the points missing, as long as the available points form a system that is solvable. Other examples produce similar graphs.

4 Related Work

Although we believe our system is novel, there are a variety of systems that can localize a device in its environment with the use of an image. Many of these are designed for robot navigation, and make use of odometry data and rely on many images to localize accurately. Early work shows that matching building features to refine location is feasible [5], but requires a highly accurate estimate of current position and orientation. More recent work makes use of robust image features such as those produced by SIFT [6]. The vSLAM system simultaneously builds a map and localizes images features within that map [7]. Other robotic navigation systems use many images along a path and visibility information to calculate likely locations [8]. None of these systems are suitable for our desired scenario where odometry data and multiple images are not available.

Image analysis systems that are not intended for localization also provide useful components. Photo Tourism can solve for 3-D locations of feature points in a scene and calculate camera positions from a large group of images [9]. New images can be matched to the current model, producing an accurate camera pose in relation to the scene. Annotations can be transferred between images by using the underlying 3-D structure. Photo Tourism relies on distinct SIFT features which most hallways lack, and it is not designed for quickly finding a camera position in a large area given a location estimate. For this reason, we may leverage the Photo Tourism system in open areas, but it is not usable for hallways. Systems to recognize landmarks in outdoor environments and provide annotations also exist [10]. This also relies on SIFT features, and does not actually generate a refined location, but merely identifies what objects might be in the image. Providing a database of geocoded features is also much higher cost than providing a building floorplan. Although these systems support similar interactions, they are not suitable for use on hallway images.

Augmented reality systems share a similar goal of information overlay. These systems tend to be object centric, and often tag objects with special markers to facilitate their location in an image [11]. Other systems actively project structured light on the environment to aide in localization [12], but would have

difficulty in a hallway environment. Existing augmented reality systems provide a variety of examples of information overlay and may be a source for applications of this system, but do not currently support hallway environments without special tagging or special hardware.

5 System Extensions

We have demonstrated the feasibility of an image based localization system for overlaying information. However, there are still many opportunities for improvement and expansion of this system.

This would not be an interesting problem if the simple methods proposed here worked for all situations. There are many cases that will cause problems for our current feature extraction method; a few of these are shown in figure 9. The biggest problem area for the simple approach presented here is in segmenting the floor. If the floor contains a large scale pattern, or the walls and floor are similar color, it will fail. For this reason we are investigating feature detectors that are tailored to the building, or potentially even different portions of the building. In addition to providing better feature detection in more environments, these specialized detectors would also provide more information, such as what type of feature (wall corner, left of door, right of door) which could be used to reduce the number of possible matches and speed up the algorithm. Our current feature detection also takes a significant portion of the runtime, and a more targeted system may be able to run faster.

To cover a wider range of environments, a mixture of floorplan-based features and image based features can be used. Although our motivating example of navigation instructions are not as useful within a room, there are applications that could use this feature. There are few features in the floorplan that are visible in an average room, however the presence of objects in these rooms provides features that are more likely distinguishable with SIFT [6] and Photo Tourism [9]. See our prior work for an explanation of how our system can be adapted to these open types of environments [2].

It is also important to consider response time in order to use this approach in an application. Although are current application is nearly fast enough, there are additional factors to consider. If all the processing is done on a server and not on the phone, the time to send the image across a network must also be considered. We assume a Wi-Fi enabled phone for the location system, and use 640x480 images, but other scenarios may take significantly more network time. It is also possible that some or all of the processing could be done on the phone as the processing power of phones continues to increase. This would allow less data to be sent across the network, reducing the time needed for transfer.

Video based "magic lens" or augmented reality type applications are also an area requiring optimization. Even if the initial camera pose takes some time to compute, assuming relatively small motion between frames it should be possible to compute an update to camera pose in a fraction of the time required for the general case. Since this system uses data in the image itself to calculate where to show things on the image, it should not be subject to the disconcerting lag or misalignment in systems that use separate sensors to determine where the camera is aimed.

A system like this would make developing and using augmented reality applications for indoor environments much easier. Environments would not need to be instrumented with additional features; at a minimum a floorplan is required in order to deploy the system, although additional information may improve robustness. Users would only need to have a camera phone to take advantage of the applications. We believe the navigation application is a compelling example that works well with this system, and thus will be our first application. However, many other types of applications may be built on this system. Timely building information, such as nearby conference room availability, could be provided by linking with a calendaring

Figure 9: Examples of images that pose problems for simple segmentation. Changes in floor material, reflective floors, and unusual structures such as catwalks break simple assumptions, but they should all be handled properly. Additionally, people or clutter in the hallways will cause problems detecting features, as will environments with low contrast difference between floors and walls.

system. More detailed building plans may allow displaying locations of electrical or plumbing lines in X-Ray vision style in order to aid service workers. We believe the design of our system will also extend to outdoor environments, allowing for a similar class of applications outdoors which can leverage the GPS infrastructure, such as tour guide systems or an easy index into information about the current surroundings.

6 Conclusion

We have presented results and analysis that demonstrate it is possible to provide the capability of information overlay on camera phones for indoor environments. In addition to leveraging existing systems that perform coarse grained localization with radio signals, we also make use of an image from the camera to refine location and camera pose. Our framework allows localization from images in existing environments with standard hardware. As demonstrated, we can obtain a highly accurate result in a short period of time, which degrades gracefully when less data is available. This platform allows for easy construction of augmented reality type applications with minimal infrastructure cost, for example, a navigation assistant to provide a low cognitive load interface by overlaying information on a user's camera phone.

7 Acknowledgements

This work is supported by NIDRR under our ACCESS project. Thanks to Noah Snavely, Steve Seitz, Linda Shapiro, and Alan Liu for their assistance on this project.

References

[1] Liu, A.L., Hile, H., Kautz, H., Borriello, G., Brown, P.A., Harniss, M., Johnson, K.: Indoor wayfinding: developing a functional interface for individuals with cognitive impairments. In: Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility, New York, NY, USA, ACM Press (2006) 95–102

[2] Hile, H., Borriello, G.: Information overlay for camera phones in indoor environments. In: Location and Context Awareness, Third International Symposium (LoCA 2007), Springer (2007) 68–84

[3] Ferris, B., Haehnel, D., Fox, D.: Gaussian processes for signal strength-based location estimation. In: Proceedings of Robotics: Science and Systems, Philadelphia, USA (August 2006)

[4] Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, IEEE Computer Society (1999)

[5] Kosaka, A., Pan, J.: Purdue experiments in model-based vision for hallway navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) **1** (1995) 87–96

[6] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2) (2004) 91–110

[7] Karlsson, N., di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.: The vslam algorithm for robust localization and mapping. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. (April 2005) 24–29

[8] Wolf, J., Burgard, W., Burkhardt, H.: Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In: Proceedings of the IEEE International Conference on Robotics and Automation. (2002)

[9] Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. **25**(3) (2006) 835–846

[10] Fritz, G., Seifert, C., Paletta, L.: A mobile vision system for urban detection with informative local descriptors. In: ICVS '06: Proceedings of the Fourth IEEE Intl Conference on Computer Vision Systems, IEEE Computer Society (2006)

[11] Hile, H., Kim, J., Borriello, G.: Microbiology tray and pipette tracking as a proactive tangible user interface. In: Pervasive Computing. Volume 3001. (2004) 323–339

[12] Köhler, M., Patel, S., Summet, J., Stuntebeck, E., Abowd, G.: Tracksense: Infrastructure free precise indoor positioning using projected patterns. In: Pervasive Computing, to appear. (2007)