A Hybrid Positioning System for Indoor Navigation on Mobile Phones using Panoramic Images

Van Vinh Nguyen and Jong Weon Lee
Department of Digital Contents, Sejong University
Seoul, South Korea (143-747)
[e-mail: vinhnv2@yahoo.com, jwlee@sejong.ac.kr]
*Corresponding author: Jong Weon Lee

Received November 6, 2011; revised January 27, 2012; accepted February 20, 2012; published March 25, 2012

Abstract

In this paper, we propose a novel positioning system for indoor navigation which helps a user navigate easily to desired destinations in an unfamiliar indoor environment using his mobile phone. The system requires only the user's mobile phone with its basic equipped sensors such as a camera and a compass. The system tracks user's positions and orientations using a vision-based approach that utilizes 360^{o} panoramic images captured in the environment. To improve the robustness of the vision-based method, we exploit a digital compass that is widely installed on modern mobile phones. This hybrid solution outperforms existing mobile phone positioning methods by reducing the error of position estimation to around 0.7 meters. In addition, to enable the proposed system working independently on mobile phone without the requirement of additional hardware or external infrastructure, we employ a modified version of a fast and robust feature matching scheme using Histogrammed Intensity Patch. The experiments show that the proposed positioning system achieves good performance while running on a mobile phone with a responding time of around 1 second.

Keywords: Vision-based positioning, indoor navigation, location-based service, natural feature matching, panorama tracking, augmented reality

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA), under the Culture Technology(CT) Research & Development Program 2011.

DOI: 10.3837/tiis.2012.03.004

1. Introduction

The location information of devices in unknown environments has become a key issue for many emerging applications and location-based services such as navigation systems. Many navigation systems have been proposed and developed for moving devices such as vehicles, robots and airplanes. Recently, due to the impressive development of mobile phones, navigation on mobile phone has received lots of attention. Generally, most of navigation services consider Global Positioning System (GPS) as the main sensor for exploring unfamiliar environments. However, since GPS signal is not available in indoor environments, many other localization methods have been proposed such as WiFi signal strength estimation [2][20][39], infrared sensors [16], RFID-based tracking [3][19], ultrasound [17][18], Bluetooth [22], GSM [23][24], etc. Each method has its own advantages and disadvantages, so there is no overall and easy solution. Considering the popularity of modern mobile phone equipped with camera and compass, we focus on a multi-sensor based positioning method for navigation applications on mobile phones in indoor environments. In particular, to minimize cost and deployment effort, we aim to develop a positioning method which can run on mobile phones independently without the requirement of additional hardware (e.g. central server) or external infrastructure.

The main problems or difficulties for mobile navigation applications are the accuracy of location estimation and the response time. Since most mobile phones contain a camera and a digital compass, we combine these sensors to propose a hybrid solution for more accurate position estimation. Our method is mainly based on vision-based localization approach. An image captured by a mobile phone is matched with reference images taken from target indoor environment and the position of the mobile phone is coarsely estimated based on the matching, while the compass is used to calculate the position more precisely in run-time.

In vision-based localization approach, a database of reference images is often required. Reference images should be taken so that they cover the targeting environment. To reduce the number of required reference images for creating the database, we utilize 360^{o} panoramic images, which can be easily captured by wide-angle cameras, fish-eye cameras or other omni-directional cameras.

For sensing mobile phone's position, reference images in the database are compared with an image captured by a mobile phone's camera to find the closest match using an image matching scheme. Because the system aims to run on mobile phones that have limitations on hardware specifications such as low CPU's speed and limited memory bandwidth, the overall performance should be fast enough so that the response time of the system is reasonable. In order to speed up the matching process, we implement a very fast matching scheme based on Histogrammed Intensity Patch (HIP) [12]. We slightly modify the HIP method to adapt with our conditions.

The remainder of this paper is structured as follows. Section 2 surveys related works in positioning techniques, up-to-date mobile phone localization methods and natural feature descriptors. Section 3 presents the HIP method with our modifications. Section 4 describes our proposed positioning system in detail. Experiment results and discussions are presented in Section 5. We conclude with a discussion of limitations and future works in Section 6.

2. Related Works

2.1 Positioning Methods

There are many solutions developed for positioning. These solutions include techniques that use WiFi, RFID, ultrasound, Bluetooth, infrared sensors, GSM, inertial sensors, optical sensors. Active Badges [16] is the first indoor positioning system which is based on infrared sensors. The badge emits a unique ID via infrared sensors regularly and the signal is received by infrared receivers equipped in the building. The later Active Badges system [17] relies on ultrasound devices deployed on mobile objects and locations within the indoor environment. Crickets [18] is another ultrasonic positioning system that has positioning accuracy of 1-2 cm. LANDMARC is proposed as a RFID positioning system [19] that contains RFID readers and tags. RFID active tag is preprogrammed with an ID to be identified by the readers. Taking the same approach of using customized beacons for localizing task, Bluetooth signals are used for localizing Bluetooth equipped mobile phones [22]. The main disadvantage of these above positioning solutions is that they require specialized infrastructure which implies pre-deployment efforts and significant cost. To avoid this problem, other solutions exploits existing infrastructure (e.g. WiFi access point) to enable indoor localization. RADAR [20] is a RF based system that use Received Signal Strength (RSS) approach with pre-WiFi WLANs for locating objects or people in indoors. The basic idea is that each location in the environment is fingerprinted with a vector of received signal strength measurements of the transmitters. A mobile device's position is estimated by matching the observed RSS vector against a database. Horus, an improved RSS-based method using a maximum likelihood approach is introduced in [21]. To enhance RSS-based positioning accuracy, a Path Loss Exponent Estimation Algorithm is proposed in [39] which only requires four beacon nodes to construct an indoor environment radio propagation loss model. GSM cell towers id and signal strength can be used for indoor localization [23][24]. An inertial method [25] has been developed where low cost gyros are used for position determination. A camera is popular choice for identifying location. Image-based positioning approach that user's location is estimated by matching scene images with a pre-built database is presented in [26][27][28]. A multi cameras system [29] is proposed to track people in the living room.

Not only single sensor is considered for positioning, integrated methods were proposed. Inertial navigation system (INS) and RFID positioning methods have been used together to calculate the position of objects [36]. RFID also can be integrated with finger printing method [37]. SurroundSense [38] uses a combination of cameras, microphone and accelerometers.

2.2 Positioning on Mobile Phone

Due to the popularity of mobile phone equipped with basic sensors and the demand from the development of location-based services, positioning on mobile phone has become a hot topic. Recently, Devin Smittle et al. [30] combine GPS receiver, accelerometer and compass, found in most mobile phones, to calculate location of users. The authors introduce an Adaptive Dead Reckoning method that can be implemented on mobile phones without requiring additional infrastructure. The average error is reported as 2.6 meters. A similar work using GPS and accelerometer for indoor localizing with mobile phone is presented in [31]. GPS is used to find the current building, while the accelerometer is used to recognize the user's dynamic activities to determine his/her location within the building. Taking different approach, a mobile phone-based indoor navigation solution is proposed in [32] where a custom inertial navigation system (INS) is used to locate the mobile phone, while the mobile phone collects additional information from its sensors to correct the position received from INS. In the work described

in [33], WiFi is employed to localizing a mobile phone. The work is stated as the first RSSI fingerprint-based application delivering up to 1.5 meters accuracy without the requirements of complex hardware. Another WiFi-based indoor localization system is presented by Krishna et al. [34]. The authors propose EZ Localization algorithm that leverages existing infrastructure without requiring any explicit pre-deployment effort (e.g. building detailed RF maps).The disadvantage of using EZ is low accuracy with a median error of 2 meters. Ultrasound is another promising technique for localization task and the ability of using a mobile phone speaker for accurate indoor positioning is investigated at the paper [35].

For vision-based positioning approach, to the best of our knowledge, only few methods are available on mobile phones so far. Mulloni et al. [4] present the indoor positioning and navigation system based on fiducial markers that are pre-designed to contain an ID number. Markers are attached to designed places and user's locations are estimated by tracking those markers. A similar approach using pre-designed markers is also presented in S. Saito and his colleagues' work [5] in which location information is encoded inside markers. Using markers for determining user's location is easy to deploy, but it locks a user to the fixed environment for sensing current locations. A user has to stand at positions that are near markers and point his phone's camera towards the marker for getting information. In other words, this approach will limit the moving area of the user. In our proposed approach, we try to make a user freely move around an environment while providing navigation guides anywhere he is staying. To enable this feature, we apply natural feature-based matching scheme for our positioning system. A fully natural feature-based indoor navigation system targeting on a camera phone has been proposed in [13], which is based on floor corners and SIFT features. This work relies on client-server architecture which outsources all time-consuming tasks to a server to reduce response time but the system still suffers from late response. Another approach which is based on a 3D point dataset to localize mobile user's 6DOF pose is introduced in [15]. In their proposed method, the sparse 3D point reconstruction is based on natural features extracted from a large set of training images. However, to generate an efficient 3D point dataset, it requires a lot of input images. In our approach presented in this paper, we employ panoramic images to reduce the number of training images while still maintain the robustness of mobile phone localization. In addition, due to navigation purpose, we focus on how to correctly determine user's position and orientation in 2D coordinates instead of user's full 6DOF pose estimation.

There are many mobile phone based navigation systems for outdoor environment that could be useful for our research. However, most of those outdoor navigation systems cannot be directly used in indoor environment to achieve results similar to our proposed system. One of the reasons is that most of current mobile phone based navigation systems for outdoor environment are based on GPS which is not available in indoor environment. A mobile navigation system is proposed in [40] which utilize built-in GPS receiver and 2D map to navigate user location in campus environment. Because GPS signals are not available in indoor environments, GPS-based navigation systems used for outdoor environment cannot be directly applied to an indoor environment. For vision-based approach, some mobile phone based systems have been proposed to work for outdoor environment. However, their methods are either not suitable to be directly used in indoor environment or not easy to use in large environments as our proposed system. An outdoor augmented reality system for mobile phone is proposed in [41]. The system matches camera-phone images against a large database of location-tagged images to recognize the target. In this system, the database is created by manually collecting images of the environment using camera phones. To cover a target environment, creating the database requires a lot of time and efforts, and it is also difficult to

deploy the system to larger environments. Besides, this system's accuracy strongly relies on database images' location information generated by mobile phone and user's location from built-in GPS receiver. Since built-in GPS sensors normally have low accuracies, the system's accuracy must be much lower than our system's desired accuracy. Hybrid approaches which utilize 3D models to track outdoor scenes and estimate user poses are presented in [42] and [43]. Those approaches rely on GPS data and other sensors to get a rough estimate of the camera pose, while 3D models are used to refine the pose. However, those systems are complicated and require external equipments. Furthermore, similar to the disadvantage of the system mentioned in [41], creating 3D models for tracking targets in large environments requires a lot of efforts and becomes a daunting task that makes those systems difficult to be used in large environments. Therefore, even through some of the outdoor systems can report having higher accuracy, they are not applicable as our proposed system in large environments. Because one of main advantages of our system is that we utilize panoramic images to generate database, thus, our database is small and requires less efforts to be generated. Other advantage is that our system only depends on mobile phone's built-in sensors and does not require any extra devices. Therefore, our system is simple to be deployed and it is easy to be applied to large environments. Besides, current vision-based approaches on mobile phone are mostly based on SIFT or SURF matching method, while our system uses HIP method [12] which is much faster than SIFT [7] or SURF [9]. Therefore, in general our system is faster in terms of matching speed compared to other outdoor existing systems.

2.3 Matching Schemes

Many approaches for matching the same objects or regions in different images have been proposed. Point-based approach uses interest point detectors and local descriptors to localize and describe image features. SIFT (scale invariant feature transform) method [7] is well-known for its robustness. The SIFT feature descriptor is invariant to scale, orientation, affine distortion and partially invariant to illumination changes. A research mentioned in [8] indicates that SIFT-based descriptors outperform other local descriptors on both textured and structured scenes. The SURF descriptor [9] is partly inspired by the SIFT descriptor and has a similar performance to the SIFT descriptor, while at the same time being much faster. However, computation requirement of those descriptors is usually expensive, thus they are not suitable for mobile phone platform. Recently, there has been some impressive work on developing new matching schemes targeting on mobile phones. Daniel et al. [10] present two techniques for natural feature tracking in real-time on mobile phones in which they modify existing feature descriptors to make them suitable for mobile phones. The Histogrammed Intensity Patch (HIP) approach is firstly introduced in [1] and a rotation-invariant version is presented in [12]. The rotation-invariant HIP-based matching scheme [12], which we apply to our system, seems to be the fastest matching scheme running on mobile phones so far. It shows comparable robustness to Daniel's work while operating about 4 times faster and requiring 5-10 times less memory. In this paper, we often mention the HIP-based matching method [12] so we refer it as "HIP method" from now on.

3. Matching Scheme

Because our matching method is strongly based on Histogrammed Intensity Patch method, we would like to give a short summarization of the method and then present our modifications. Basically, the HIP method has two main phases: a training phase and a run-time matching phase.

3.1 Training Features

In the training phase, a set of training images covering the entire environment is used for generating a database of features. The set is divided into several different viewpoint bins. Each viewpoint bin covers a small range of scale and affine viewpoint parameters. In practice, training images are artificially generated by warping a single reference image. The training process has two stages: First, every training image in a viewpoint bin is processed by running FAST corner detector [6], and then sub-features with position and orientation information are extracted based on patches extracted around the corresponding detected corners. The patch is a square 8x8 sampling grid centred on the interest point, with a 1-pixel gap between samples. Next, sub-features are clustered to identify repeatable features. The most repetitively detected features will be considered as HIP features and be added to a database.

3.2 HIP Feature Representation

A HIP feature is described based on the combination of quantized patches from its related sub-features. The HIP feature contains 64 independent histograms of 5 quantized intensity levels. In particular, the feature stored in the database can be presented as:

$$\begin{matrix} D_{0,0} & D_{0,1} & D_{0,2} & D_{0,3} & D_{0,4} \\ D_{1,0} & D_{1,1} & D_{1,2} & D_{1,3} & D_{1,4} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{63,0} & D_{63,1} & D_{63,2} & D_{63,3} & D_{63,4}, \end{matrix} \tag{1}$$

where each row corresponds to the quantized histogram for a single pixel of the HIP model, and:

$$D_{i,j} = \begin{cases} 1 & if\ P(B_j < I(x_i, y_i) < B_{j+1}) \quad < 0.05 \\ 0 & otherwise \end{cases} \tag{2}$$

where B_j is the minimum intensity value of a histogram bin j and I(x_i, y_i) is the normalized value of pixel i.

3.3 Runtime Matching

Runtime captured images are processed to get patches. Runtime patches are extracted from a rotated sparse sampling grid and then quantized into bitwise descriptions as in the training phase but have a small difference in the binarization step as the following rule:

$$R_{i,j} = \begin{cases} 1 & if\ B_j < RP(x_i, y_j) < B_{j+1} \\ 0 & otherwise \end{cases} \tag{3}$$

where $RP(x_i, y_i)$ is the value of pixel i in the normalized runtime patch.

To measure the similarity of a runtime patch with one in database, a dissimilarity score is computed by counting the number of bits where both $D_{i,j}$ and $R_{i,j}$ have a value of 1:

$$e = \sum_{i,j} D_{i,j} \otimes R_{i,j} \tag{4}$$

Two features are considered as a match if their patches' dissimilarity score is below a threshold. Because the error measurement can be calculated by bitwise (AND and OR) operations, the matching process would be very fast. Furthermore, the authors suggest an indexing scheme and a tree-based lookup scheme to reduce time of matching process while dealing with large number of features in a database.

3.4 Modified Matching Method

In the original HIP-based matching scheme, the authors use 9 scales for training a target. The number of scales is to ensure that the target is still tracked correctly when the camera moves far from the target. However, in our situation, since we build a feature database of indoor environment using panoramic images which are taken at the center positions of featured places, we often face a zoom-in problem when a user come closer to a target rather than a zoom-out situation. It means that captured images of an object in runtime are often bigger than training images of the same object. Therefore, in training phase, we reduce number of scales to 5 that will not affect the system's robustness while decreasing the number of features stored in database, thus, increasing matching speed. Moreover, in order to solve the zoom-in problem more effectively, we modify the runtime matching phase by adding multi-scaling scheme: for each runtime image, we create several sub-images by scaling down the original one. Practically, we generate 4 sub-images and the reducing factor is 0.7. Features of these scaled images are extracted and grouped as one set of features of the runtime image. This set will be used for matching the image with the database. In case of zoom-in, matching often fails because no training image has the same scale as captured image. Runtime scaling method generates smaller sub-images that have the similar size compared to training images, so features of those sub-images are similar to features in the database.

4. The Proposed Positioning System

4.1 System Overview

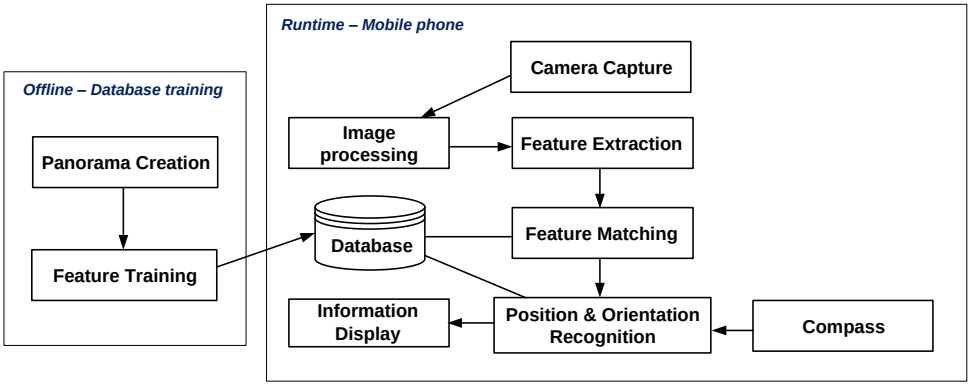


Fig. 1. System's workflow diagram.

The proposed system is structured as in **Fig. 1**. In the training phase, a database of reference images is created based on panoramic images captured in a targeting indoor environment. Each reference image is described in the database as a set of features that are created from the feature training process. During runtime, a captured image from mobile phone's camera is processed to extract features, and then a matching scheme is used to compare the runtime captured image with ones in the database. If any match is found, mobile device's position and orientation are estimated based on the match and compass orientation.

4.2 Database Creation

In our system, a database of reference images is required for runtime matching. The large set

of reference images should be taken so that they cover the targeting environment. For minimizing the number of required images, we use panoramic images. A database is created by following four steps:

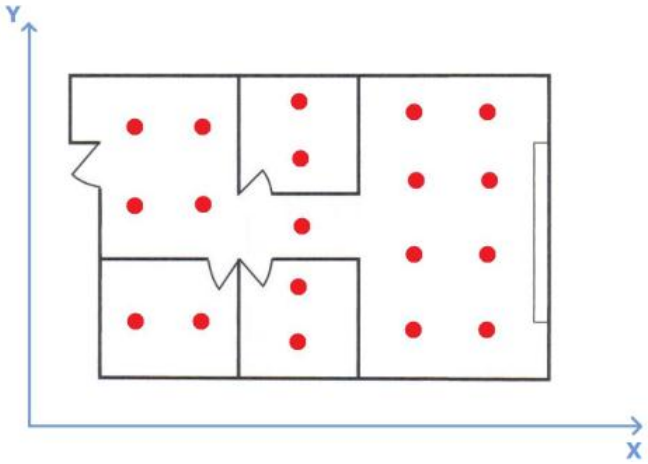


Fig. 2. An example of an indoor map with 2D reference coordinate systems. Red marks are positions of a camera capturing panoramic images.

Step 1. We setup a 2D reference coordinate system in the targeting indoor environment and then coordinates of featured places (e.g. rooms in a building) are roughly estimated.
Step 2. For each featured place, we create several 360^o panoramic images. The number of panoramic images has to be minimized to reduce the matching time while maintaining the required resolution of position estimation. Practically, we divide each place into 3 x 3 m cells and take one panoramic image in the center of each cell. The position and orientation are recorded with each panoramic image. The orientation of panoramic image means the angle between the view's center and Y-axis of the reference coordinate system.
Step 3. We divide horizontally each panoramic image into four sub-images. That means one sub-image covers 90^o of the view and it is easy to calculate the orientation of each sub-image based on the orientation of the original panoramic image.
Step 4. Finally, we run HIP feature training on each sub-image and then store HIP features with corresponding positions and orientations of the sub-image into the database.

Note that position and orientation of any sub-image in the database mean the position and orientation of the capturing camera (or the view). Also, in this paper the term "sub-image" is same as the "reference image."

4.3 Panorama Issues

There are several ways to capture panoramic images. One of the common methods is stitching photos of the same scene to get a panorama. This is easy because we only need to get perspective photos of a scene by a normal digital camera and then use a stitching application to generate panoramic images. The disadvantage of this method is that if the number of panoramas needed for creating a database is big, the number of perspective photos is also significantly increased. This requires lots of capturing effort. Besides, determining the orientation of a stitching panorama is not easy. In order to reduce efforts for a database setup, we use fish-eye camera lens for capturing scene. A 360^o panoramic image can be created by

merging several wide-angle images.

In fact, we can use any kind of camera to capture the scene and then generate panoramas. Howerver in this paper, we are focusing on fisheye panorama images because there are existing fisheye panorama images of outdoor scene which are available and be used for "street view" mode of some popular online map applications like Daum map [44]. Since those images covers large areas such as cities, or countries, if we can successfully develop a positioning system based on fisheye panorama images, it is easy to practically extend our system to large outdoor environment without efforts of capturing and generating database images.

Fig. 3. Example of correcting fisheye distortion for a training image.

A problem of using panorama captured by the fisheye lens is fisheye distortion occur in captured images which may affect the performance of the matching phase. To solve this problem, we divide a panoramic image into 4 sub-images so that position of the fisheye distortion's centric point is almost same as center position of sub-images. After that, we remove the distortion of sub-images as shown in **Fig. 3**.

4.4 Ineffective Point Removal

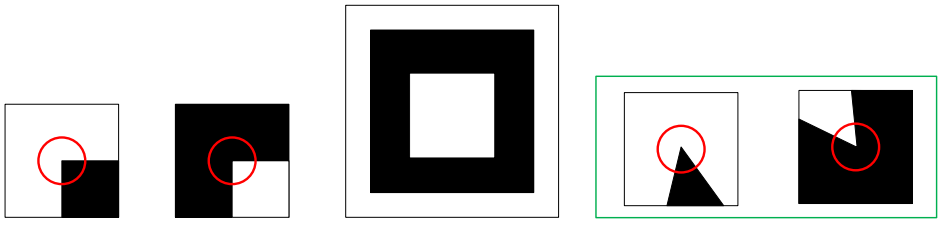


Fig. 4. a) "White perfect" corner. b) "Black perfect" corner. c) A template image for "perfect corner" removal. d) Examples of remaining corners after the removal process.

In indoor environment, we found that there are two types of corners which are frequently seen. They are "white perfect corner" and "black perfect corner" as shown in **Fig. 4**. These two corners can be easily detected in an image of any rectangular object such as a frame, a table or a door. Since FAST detector [6] recognizes these two corners as FAST corners, the database contains a lot of these corners. However, because these two corners are commonly found in different objects in the environment, they increase the number of outliers in matching, thus, reduce the matching's performance. To solve this problem, we eliminate these types of corner

in the database by using a template image shown in **Fig. 4**. Any feature in the database which matches with HIP features extracted from the template image will be removed. After the removal process, the database contains corners which should have "*centered angle*" either less than or greater than 90^o as seen in **Fig.4-d**.

4.5 Update 2D-features

Since we take one panoramic image on every 3x3 meters cell, neighbouring panoramas often have overlap areas. We would like to take an advantage of those overlaps for estimating user phone's position more precisely.

We compare each sub-image in the database with sub-images of a neighbouring panorama. If any match is found between two images, 2D reference coordinates of matched features are calculated and then updated into the database. The calculation of a matched feature's reference-coordinates can refer to the following theory:

Assuming that every sub-image in the database is applied distortion removal, it can be considered as a perspective image which is captured by a virtual perspective camera. As each sub-image covers 90^o in horizontal, the virtual perspective camera should have 90^o of horizontal field of view that is named as "*HAoV*". Also, position and orientation of each sub-image are stored in the database. Therefore, for each sub-image we can simulate a corresponding virtual camera's view in 2D reference coordinates. **Fig. 5** shows a simulation of the virtual camera's position and orientation of two reference images. The first view is placed at the position O_1 and the second view is at position O_2. "*D*" is horizontal camera sensor size of the virtual camera, while "*f*" is the effective focal length.

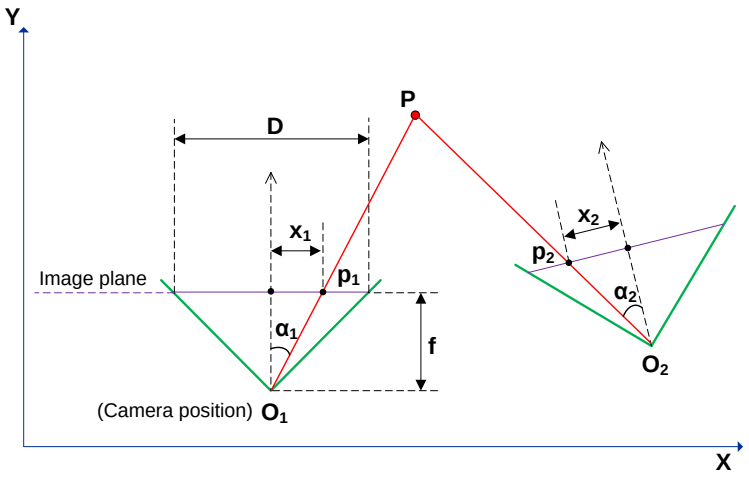


Fig. 5. Simulation of camera views of 2 sub-images in 2D reference coordinates

Assuming that these images have overlap areas which are detected by the matching scheme. There is at least one matched point-pair: Point "p_1" in the first image is matched with an image-point "p_2" in the second one. According to the "perspective camera model" theory, we know that "p_1" and "p_2" are the outcome of the perspective projection from a real 2D-point "P" to two image planes, respectively. It means "P" is the intersection point of a line "O_1-p_1" and a line "O_2-p_2". To calculate 2D reference coordinates of "P", we only need to determine two angles α_1 and α_2 because position and orientation of two camera's view are known.

From the perspective project mode, we have

$$HAoV = 2\arctan\left(\frac{D}{2f}\right) \qquad or \qquad D = 2f\tan\left(\frac{HAoV}{2}\right) \tag{5}$$

and

$$\alpha_1 = \arctan\left(\frac{x_1}{f}\right) = \arctan\left(\frac{x_1 * D}{f * D}\right) \tag{6}$$

Let's assume that

$$r_1 = \frac{x_1}{D} \qquad or \qquad x_1 = r_1 D \tag{7}$$

Formula (6) becomes:

$$\alpha_1 = \arctan\left(r_1\frac{D}{f}\right) = \arctan\left(2r_1\tan\left(\frac{HAoV}{2}\right)\right) \tag{8}$$

Because $HAoV = 90^o$, then

$$\alpha_1 = \arctan\left(2r_1\tan\left(45^o\right)\right) = \arctan\left(2r_1\right) \tag{9}$$

Originally, "D" and x_1 are in mm. Since r_1 is a ratio measurement, it can be calculated in pixels dimension instead. At this point, x_1 becomes a horizontal distance (in pixels) from p_1 to the center of the image and "D" can be considered as the width of the image.
Similarly, we can calculate the value of α_2 as

$$\alpha_2 = \arctan\left(\frac{x_2}{f}\right) = \arctan\left(2r_2\right) \tag{10}$$

where

$$r_2 = \frac{x_2}{D} \tag{11}$$

Having 2 angles α_1 and α_2, we can find the equations of two lines: "O_1-p_1" and "O_2-p_2". Then, "P" is determined as the intersection of lines.

After this step of 2D-features update, the database contains both normal features and 2D-features which have reference coordinates. The number of 2D-features of a reference image depends on the number of its neighbor images and the sizes of their overlap regions.

4.6 Position and Orientation Estimation

Each runtime captured image is compared to the database. If any match is found, the position and orientation of mobile phone is calculated. At this point, there are two possible situations that can happen.

Situation 1: The captured image is matched but no matched feature in the database has 2D reference coordinates.

In this case, corresponding position and orientation of the matched image in the database are considered as the position and orientation of the mobile phone. However, the captured image from the mobile phone is usually matched as a region of reference images as demonstrated in **Fig. 6**. Moreover, these images are not often vertically aligned. Therefore, we need to re-calculate the real orientation of the phone based on orientation of the corresponding panoramic image. To do that, we rotate the captured image so that it is vertically aligned with the reference image, and then we apply homography method to estimate the angle difference (or error) between real orientation of mobile phone and orientation of the reference image.

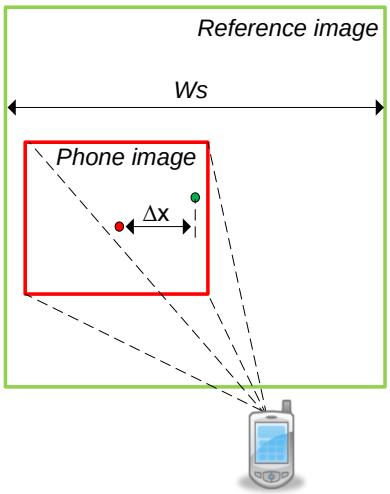


Fig. 6. Mobile phone orientation error calculation.

To simplify the calculation, we use RANSAC[11] to estimate the central point and four corners of the matched reference image in the captured image's frame. Having that information, we calculate the difference of 2 central points' x-coordinates in captured image's frame. Moreover, since the reference image covers 90^o of the view, we can estimate the real mobile phone's orientaion based on the difference angle which is calculated using the following equation:

$$\Delta\beta = \frac{\Delta x * 90^o}{Ws} \tag{12}$$

where Ws is the width of reference image in captured image's frame and Δx is a difference of 2 central points' x-coordinate in captured image's frame.

Situation 2: The captured image is matched and there are at least 2 matched features in the database that have 2D reference coordinates.

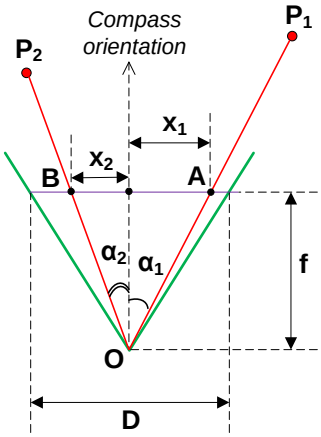


Fig. 7. Mobile phone's position estimation using compass.

In this case, we use two 2D-features and compass's orientation to get mobile phone's position. Assuming that two features are "A" and "B" which have 2D coordinates as "P_1" and "P_2", respectively. As seen in **Fig. 7**, to estimate mobile phone's position "O", we need to find line equations of two lines "P_1A" and "P_2B". Because coordinates of P_1 and P_2 are known, to get the equations, we only need to know the angles of two lines with X-axis of the reference coordinates. Besides, camera phone's orientation is known using the digital compass attached to the phone, to calculate two lines' angles, we only need to determine two angles α_1 and α_2 which are demonstrated in **Fig. 7**.

Considering the captured image is perspective image, we have an equation to calculate α_1 as same as equation 8 in section 4.5:

$$r_1 = \frac{x_1}{D} \tag{13}$$

$$\alpha_1 = \arctan\left(r_1 \frac{D}{f}\right) = \arctan\left(2r_1 \tan\left(\frac{HAoV}{2}\right)\right) \tag{14}$$

where *HAoV*, "*D*", "*f*" are respectively the horizontal field of view, a camera sensor size and the focal length of the mobile phone's camera. Similarly, we can calculate α_2 as

$$r_2 = \frac{x_2}{D} \tag{15}$$

$$\alpha_2 = \arctan\left(2r_2 \tan\left(\frac{HAoV}{2}\right)\right) \tag{16}$$

Having 2 angles α_1 and α_2, we can caculate angles of two lines with X-axis and then we can find the equations of two lines: "P_1A" and "P_2B". Finally, "O" is determined as the intersection of two lines.

Be noted that if there are more than 2D-features, each pair of 2D-features will be used to calculate the phone's position. The final phone's position is considered as the center of the set of estimated ones.

However, in fact the compass's orientation always has some errors compared to real orientation due to interferences from surrounding metallic objects. To get correct orientation of the phone, we iteratively change the orientation by increasing and decreasing it within a threshold value and then re-calculate phone's positions. The final chosen orientation is the one which generates the most convergent set of estimated positions.

Note that a runtime image can be matched with several images in database. However, the position estimation usually generates better results when the runtime captured image is matched with the nearest panoramic image than other ones. The nearest panoramic image is selected by comparing the horizontal angles of matched areas between the runtime captured image and reference panoramic images. Since cameras used for reference images and the runtime captured image are different, this field-of-view difference is compensated before the comparison. After the compensation step, the panoramic image that has the closest horizontal angles of the matched area is selected as one captured at the location closest to the position of the runtime image.

4.7 Position Tracking

To reduce matching time in runtime, a previous position tracking mechanism is employed. Each captured image is first compared to reference images which have positions (or coordinates) close to the previous position. Assuming a user is moving slowly, this mechanism

can avoid comparing a captured image to all reference images, which is a significant time consuming process.

5. Experimental Results

For performance evaluation, at first we implemented our modified HIP method and then compared its performance with SURF. The experiment shows that compared to SURF, our matching method has similar correct matching ratio, while much faster than SURF. Next, we have implemented and tested the proposed positioning system on an iPhone 3GS. The testing environment was a floor of a museum that has 5 rooms as shown in **Fig. 8**. We captured the museum using a DSLR camera with a fisheye lens. Although vertical field of view of the lens is nearly 180^o, we created panoramic images which cover the scene 90^o in vertical and 360^o in horizontal for navigation purpose. The reason for using only 90-degrees panoramas is that a mobile phone's camera has small field of view and the camera should has forward orientation in navigation mode. 90-degree in vertical of reference images is enough to contain good features of the scene for position estimation in runtime, while eliminating unnecessary features on ceilings or floors.

Fig. 8. A map of the tested museum floor and an example panorama taken in one room.

5.1 Matching on Mobile Phone

We implemented and tested the modified version of HIP method on the iPhone. The speed of wide baseline matching on the iPhone is about 9 fps.

Another test was taken to evaluate the usefulness of ineffective point remove (IPR) step which is described in section 4.4. In this test, we use a large set of images of rectangular objects seen in indoor environments such as picture frames, windows, computer screens, or black boards. The experiment proves that using IPR in HIP method will increase 10% of matching percentage in indoor environment.

5.2 Correctness of Position Estimation

First, the procedure of choosing the nearest panoramic image has been tested with 400 images captured in a floor of the museum as shown in **Fig. 8**. The nearest panoramic reference image has been detected for most of the tested images.

Next, we take an evaluation on the correctness of position estimation for a mobile phone. In this experiment, 75 panoramic images captured from 5 rooms in the museum floor are used to create database. Reference images are in 800x800 pixels and there are average 1,500 HIP features extracted from a reference image. In the "2D-features update" step of database training phase, for each reference in database, we found about 2 to 6 neighboring reference images having overlapping areas to compute 2D-features. For one image, around 180 features are updated with 2D-coordinates that take 12% in total number of features.

In runtime test, we took the iPhone to go around the museum and captured the scene in pre-measured positions. 60 positions were examined and estimated coordinates were then compared to the corresponding real coordinates which are considered as ground-truth data for calculating errors. **Fig. 9** depicts the cumulative distribution of localization errors obtained from the test.

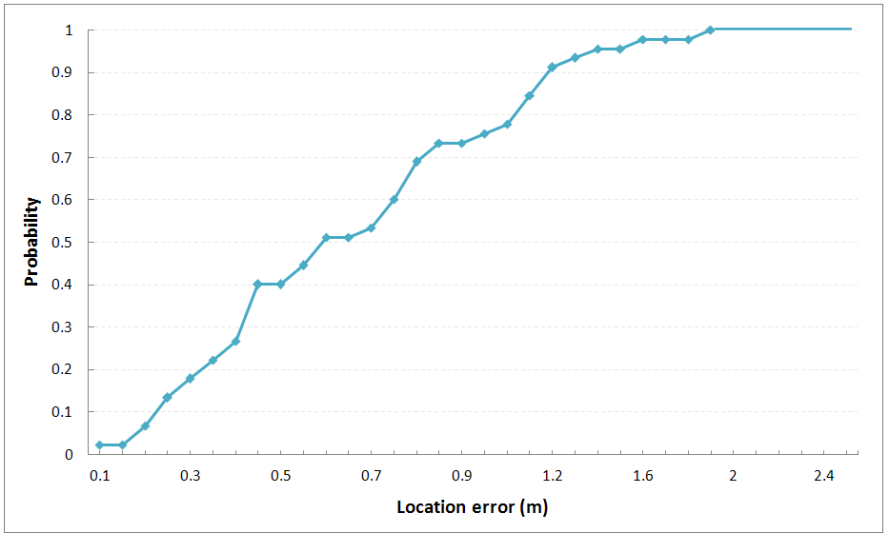


Fig. 9. CDF of localization errors in the first test

In summary, the mean error of our method is 0.68 m and the standard deviation of error is 0.40 m. This is a promising result despite the fact that, 2D-features are found and matched in only 34 tested positions. In 26 remaining positions, "*situation 1*" mentioned in section 4.6 occurred: none 2D-feature is found in the matched reference panorama. This problem often happens in positions within a texture-less area or narrow area (*e.g.* lobby) because in such these areas, there is not enough matched neighboring panoramas to generate 2D-features.

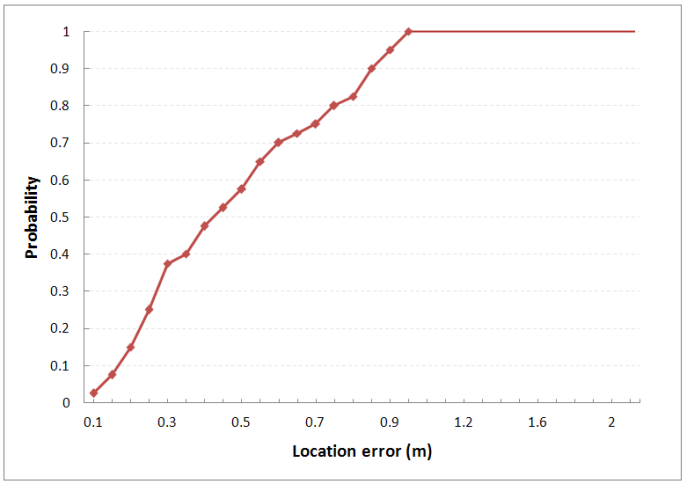


Fig.10. CDF of localization errors in the second test

We took the second experiment to estimate the robustness of our hybrid method in an ideal scenario: 2D-features are always found in any match. In this experiment, we chose a large texture room and 40 positions within this room were checked. The results came out that our method gains higher accuracy in this case. The mean error is 0.46 m and the standard deviation of error is 0.26 m. The cumulative distribution of errors is drawn in **Fig.10**.

The second experiment means that if our positioning method is applied in texture environments such as galleries or exhibition rooms, the position estimation's accuracy could be higher than the accuracy in our tested museum.

5.3 System Performance Evaluation

Having 75 panoramas captured from a room, we created a database of 300 reference images. Since the big number of reference images can increase the responding time if a runtime captured image is required to match with all of database images. To ease this problem, for each reference image in database, we records its neighbors so that when we need to find all matches in database of a runtime image, we just need to search the first matched reference image and then compare to its neighbors to find other matches. The proposed positioning system is then tested in the museum with a following scenario:

A user taking an iPhone travels around the room. When he needs to move to another place (e.g. another room), he uses the iPhone to capture scenes (or objects) near his current standing position. Based on the captured image, the system running on the iPhone will show a virtual arrow indicating the direction to the destination. Moreover, an estimated distance to the destination is also displayed.

The testing results show that the system can navigate successfully a user to a destination with response time less than 1 second in tracking mode.

In our test, matching a captured image with 300 reference images in database pays at the cost of high user's waiting time (it takes around 10 seconds in the worst case). But this problem happens only when system starts or when the position tracking fails. This is known as initialization (or re-initialization) problem which is a common problem in wide-area localization systems: the captured image has to be compared with all database images and this process will take long time if the number of reference images is big. To cope with this problem, researchers apply some possible solutions. Vocabulary tree [14] can be used for sub-linear

searching the most similar images stored in a large database. Alternatively, we can exploit others sensor signals such as WiFi, RFID, and Bluetooth for getting a coarse position (e.g room-level position), thus reducing the number of reference images needed to be compared at initialization stage.

5.4 Comparison

Table 1. List of existing positioning methods with their accuracies

Methods	Accuracy		Signals
	Mean error (m)	**Std. Dev of error (m)**	
Adaptive dead reckoning [30]	2.60	*Not reported*	GPS + accelerometer + compass
User activity modeling [31]	Floor-level		GPS + accelerometer
Fingerprinting [33]	1.5	-	WiFi
EZ Localization [34]	2.0	-	WiFi
Our method	**0.68**	**0.4**	**Camera phone + compass**

We compare our method to other existing positioning methods which have been studied in related works section. We choose the most recent positioning methods which aim to work on mobile phones and their accuracies are experimentally reported. As seen in **Table 1** that lists all methods with their accuracies, our method performs best with the lowest mean error.

6. Conclusions and Future Works

We have proposed a novel positioning solution for indoor navigation on mobile phones using panoramic images. The main contribution of our paper is that the proposed system combines two consumer-grade sensors of modern mobile phones and utilizes panoramic images to localize users in indoors without the requirements of extra hardware or external infrastructure. Experimental results show that our solution outperforms other existing methods working on mobile phones.

Since panoramas are easily created using a wide-angle camera lens, we believe that by utilizing panoramic images for creating databases, our proposed solution is applicable to large indoor environments or even to outdoors. Besides, because our positioning solution is strongly based on an image-based matching scheme, it is not only suitable for indoor navigation services, but also other location-based service using camera phone such as Augmented Reality-based applications.

In this paper, we have not considered any specific method for dealing with initialization problem because we focused on proposing and evaluating a new hybrid solution for mobile phone localization based on panoramic images. Therefore, initialization evaluation and outdoors localization on mobile phones will be our future works.

References

[1] S. Taylor, E. Rosten, and T.W.Drummond, "Robust feature matching in 2.3 microseconds," *IEEE CVPR Workshop on Feature Detectors and Descriptors*, Jun.2009. Article (CrossRef Link)

[2] Sinan Gezici, "A survey on wireless position estimation," *Wireless Personal Communications: An International Journal*, vol.44, no.3, pp.263-282, Feb.2008. Article (CrossRef Link)

[3] M. Bouet, and A.L. dos Santos, "RFID tags: Positioning principles and localization techniques," *IFIP Wireless Days, 2nd International Home Networking Conference*, Nov.2008. Article (CrossRef Link)

[4] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, "Indoor Positioning and Navigation with Camera Phones," *IEEE Pervasive Computing*, 2009. Article (CrossRef Link)

[5] S. Saito, A. Hiyama, T. Tanikawa, and M. Hirose, "Indoor Marker-based localization using coded seamless pattern for interior decoration," *Virtual Reality Conference*, 2007. Article (CrossRef Link)

[6] E. Rosten and T. Drummond, "Machine learning for high speed corner detection," In *Proc. of 9th Euproean Conference on Computer Vision*, vol.1, pp.430-443, Apr.2006. Article (CrossRef Link)

[7] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp.91-110, 2004. Article (CrossRef Link)

[8] K. Mikolajczyk, and C. Schmid, "A performance evaluation of local descriptors," In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. Article (CrossRef Link)

[9] H. Bay, T. Tuyelaars, and L. Van Gool, "Speed-up robust features," *ECCV*, pp.404-417, 2006. Article (CrossRef Link)

[10] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," *7th IEEE/ACM International Symposium on Mixed and Augmented Reality,* 2008. Article (CrossRef Link)

[11] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communication of the ACM*, vol.24, no.6, pp.381–395, 1981. Article (CrossRef Link)

[12] S. Taylor and T.W. Drummond, "Multiple target localisation at over 100 FPS," In *British Machine Vision Conference*, Sept.2009. Article (CrossRef Link)

[13] H. Hile and G. Borriello, "Information overlay for camera phones in indoor environments," In *Location- and Context-Awareness*, *Third International Symposium*, vol.4718, pp.68–84, 2007. Article (CrossRef Link)

[14] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," In *Proc. of Conference on Computer Vision and Pattern Recognition*, pp.2161-2168, 2006. Article (CrossRef Link)

[15] C. Arth, D. Wagner, M. Klopschitz, A. Irschara and D. Schmalstieg, "Wide area localization on mobile phones," *International Symposium on Mixed and Augmented Reality*, 2009. Article (CrossRef Link)

[16] R. Want, A. Hopper, V. Falcao and J. Gibbons, "The active badge location system," *ACM Transactions on Information systems*, vol.40, no.1, pp.91-102, Jan.1992. Article (CrossRef Link)

[17] A. Ward, A. Jones and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol.4, no.5, pp.42–47, 1997. Article (CrossRef Link)

[18] N.B. Priyantha, "The cricket indoor location system," *PhD Thesis*, pp.199, Jun.2005. Article (CrossRef Link)

[19] L. Ni, Y. Liu, C. Yiu and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," In *Wireles Networks entitled Pervasive Computnig and Communications*, 2004. Article (CrossRef Link)

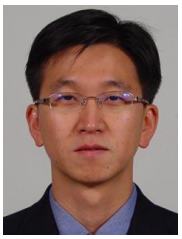
[20] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in *Proc. of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000. Article (CrossRef Link)

[21] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," *in Proc. of the 3rd international conference on Mobile systems, applications, and services*, 2005. Article (CrossRef Link)

[22] S. Hay and R. Harle, "Bluetooth tracking without discoverability," in *Proc. of 4th International Symposium on Location and Context Awareness*, 2009. Article (CrossRef Link)
[23] V. Otsason, A. Varshavsky, A. LaMarca, E. de Lara, "Accurate gsm indoor localization," pp. 141–58, 2005. Article (CrossRef Link)
[24] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, "GSM indoor localization," *Pervasive and Mobile Computing,* vol.3, no.6, pp.698–720, 2007. Article (CrossRef Link)
[25] Jussi Collin, Oleg Mezentsev and Gérard Lachapelle, "Indoor positioning system using accelerometry and high accuracy heading sensors," in *Proc. of GPS/GNSS 2003 Conference*, Sept.2003. Article (CrossRef Link)
[26] R. Elias and A. Elnahas, "An accurate indoor localization technique using image matching," *Intelligent Envionments*, 2007. Article (CrossRef Link)
[27] H. Bay, "Interactive museumgGuide," in *Proc. of UBICOMP*, Sept.2005. Article (CrossRef Link)
[28] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Itode, "Indoor localization using camera phones," in *Proc. of IEEE Worshop on Mobile Computing Systems and Applications*, 2006. Article (CrossRef Link)
[29] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale and Steve Shafer, "Multi camera Multi-person tracking easy living," *Third IEEE International Workshop on Visual Surveillance*, 2000. Article (CrossRef Link)
[30] Devin Smittle and Veselin, "Indoor localization on mobile phone platforms using adaptive dead reckoning," *MU Summer Undergraduate Research and Creative Achievements Forum*, Jul. 29, 2010. Article (CrossRef Link)
[31] Avinash Parnandi, Ken Le, Pradeep Vaghela, Aalaya Kolli, Karthik Dantu, Sameera Poduri and Gaurav S. Sukhatme, "Coarse In-Building Localization with Smartphones," *MobiCASE*, 2009. Article (CrossRef Link)
[32] C. Lukianto, C. Hönniger and H. Sternberg, "Pedestrian smartphone based indoor navigation using ultra-portable sensory equipment," *International Conference on Indoor Positioning and Indoor Navigation* , 2010. Article (CrossRef Link)
[33] E. Martin, O. Vinyals, G. Friedland and R. Bajcsy, "Precise indoor localization using smart phones," in *Proc. of ACM Multimedia*, 2010. Article (CrossRef Link)
[34] C. Krishna, P. Anand and N.P. Venkata, "Indoor localization without the pain", in *Proc. of the sixteenth annual international conference on Mobile computing and networking*, 2010. Article (CrossRef Link)
[35] Viacheslav Filonenko, Charlie Cullen, James D. Carswell, "Investigating ultrasonic positioning on mobile phones," *International Conference on Indoor Positioning and Indoor Navigation*, 2010. Article (CrossRef Link)
[36] M. Zhu, "Novel positioning algorithms for RFID-assisted 2D MEMS INS systems," in *Proc. of the Institute of Navigation*.
[37] G. Retscher and Q. Fu, "Using active RFID for positioning in navigation systems," in *Proc. of the 4th International Symposium on Location Based Services and Telecartography*, 2007.
[38] M. Azizyan, I. Constandache and R.R. Choudhury, "Surroundsense: Mobile phone localization via ambience fingerprinting," in *Proc. of ACM International Conference on Mobile Computing and Networking*, 2009. Article (CrossRef Link)
[39] Yu-Sheng Lu, Chin-Feng Lai, Chia-Cheng Hu, Yueh-Min Huang and Xiao-Hu Ge, "Path loss exponent estimation for indoor wireless sensor positioning," *KSII Transactions on Internet and Information Systems*, vol.4, no.3, pp.243-257, Jun.2010. Article (CrossRef Link)
[40] T. Mantoro, S. A. Saharuddin,and S. Selamat, "3D interactive mobile navigator structure and 2D map in campus environment using GPS," in *Proc. of the 7th International Conference on Advances in Mobile Computing and Multimedia,* Dec.2009. Article (CrossRef Link)
[41] Gabriel Takacs, Vijay Chandrasekhar, B. Girod et. al., "Outdoors augmented reality on mobile phone using Loxel based visual feature organization," in *Proc. of ACM Multimedia Information Retrieval*, Oct.2008. Article (CrossRef Link)

[42] G. Reitmayr and T. W. Drummond, "Going Out: Robust model based tracking for outdoor AR," in *Proc. of IEEE/ACM International Symposium of Mixed and Augmented Reality,* pp.109–118, 2006. Article (CrossRef Link)
[43] J. Karlekar, S. Zhou, W. Lu, Z. C. Loh, Y. Nakayama, and D. Hii, "Positioning, tracking and mapping for outdoor augmentation," in *Proc. of IEEE/ACM International Symposium of Mixed and Augmented Reality,* pp.175–184, 2010. Article (CrossRef Link)
[44] http://map.daum.net

Van Vinh Nguyen is a Ph.D. candidate in Department of Digital Contents at Sejong University, in Seoul, Korea. He received the B.S. degree in Computer Science from Hanoi University of Technology, Vietnam in 2004, and the M.S. degree in Computer Engineering from University of Ulsan, Korea in 2007. His research interests include augmented reality, real-time object detection and tracking, and vision-based positioning.



Jong Weon Lee was born in 1966. He received the B.S. degree in Electrical Engineering from Ohio University, Ohio in 1989, the M.S. degree in Electrical and Computer Engineering from University of Wisconsin at Madison, Madison in 1991, and Ph.D. degree in Computer Science from University of Southern California in 2002. He is presently Professor of Department of Digital Contents at Sejong University where his research interests include augmented reality, human-computer interaction, and serious game.