A Visual-based Positioning System

Cheng-Sheng Chien[1] and Min-Te Sun[1]
bravesheng@gmail.com, msun@csie.ncu.edu.tw

Department of Computer Science and Information Engineering,
National Central University, Taiwan

Abstract. We implemented a visual-based positioning system on the Android-based smartphone. This system uses only build-in camera and sensors in an Android phone. The system works with no other assisted equipments, so it is low-cost and very easy to deploy to consumers with a mobile device. In addition, we conducted small, median, and large-scale field experiments to examine the accuracy and stability of our system. The results indicate that our visual-based positioning system is able to provide a much better accuracy in small and median-scale experiments compared with the global positioning system. In large-scale experiments, the system is able to provide enough accuracy for applications such as virtual augmentations and location-based services.

Keywords: positioning, location-based service, LBS, android, augmented reality

1 Introduction

In recent years, the market share of smartphones has increased rapidly. We have been surrounded by the computational power as well as a variety of sensors of mobile devices. As a result, a number of novel applications and location-based services have been developed. People have gradually found the limitations of the global positioning system (GPS) and developed all kinds of different indoor positioning systems on top of Wi-Fi [1], Bluetooth [2], light digital pulse [3], and magnetic fields inside a building [4]. In addition to these approaches, some visual-based positioning systems [5] [6] have also been proposed.

In this research, we implemented a visual-based positioning system on Android-based smartphone. Our system makes use of the built-in camera and sensors to compute the relative distance and angle between the reference point and the camera. The system runs on a standalone Android smartphone with no assisted devices. It has a very low cost and is very easy to deploy to consumers who have a mobile device. The experiment results show that our proposed visual-based positioning system is both accurate and stable for applications such as indoor positioning navigation, virtual augmentations, and location-based services.

2 Related Work

In this section, we present the related work. In Section 2.1, we review the existing visual-based positioning systems. Since some of the parameters in our system also rely on filter algorithms for fine tuning, in Section 2.2, we briefly review these filter algorithms.

2.1 Existing Visual-based Positioning Systems

In 2009, Google presented Google Goggles [7]. This web service allows a user to take pictures by a smartphone, recognizes famous landmarks in pictures, and searches websites related to the landmarks automatically. In [8], the authors published an extensive landmark dataset and presented a comprehensive pipeline algorithm for mobile location recognition. These services focus on automatic landmark identification instead of positioning.

In [9], the authors developed a system that processes a cell-phone camera image and matches detected landmarks from the image to a building. The system calculates camera location and dynamically overlays the derived the characteristics of landmarks directly on the cell-phone image. This system accomplishes positioning by the extraction and matching of the environmental characteristics and features without the uses of reference objects. As a result, this system requires a huge image database to work properly.

In [6, 5], the authors developed a low-cost method of real-time positioning using the built-in camera of a cell-phone. The location of the user is determined by detecting unobtrusive known fiduciary markers around a building. Indoor navigation is allowed by the continuous scanning of environment in real time (15 Hz or more) in search of strategically placed fiduciary markers. The system is only based on paper markers (square markers or frame markers) and static digital maps. No additional infrastructure is needed.

Following these concepts, we intend to implement a real-time visual-based positioning system on an Android-based device to achieve accurate positioning.

2.2 Filter Algorithms

The readings of accelerometer and magnetometer sensors should be enough to calculate the orientation of device. However, the output of both types of sensors may be inaccurate, especially the output from the magnetometer sensors which includes a lot of noise. In order to reduce noise, we use the following two filters.

Simple Moving Average Filter In science and engineering, the simple moving average (SMA) [10] is normally taken from an equal number of data on either side of a central value. This ensures that variations in the mean are aligned with the variations in the data rather than being shifted in time.

An example of a simple equally weighted running mean for a n-day sample of closing price is the mean of the previous n days' closing prices. If those prices are Pm,Pm-1,...,Pm-(n-1) then the formula is

$$SMA = \frac{p_m + p_{m-1} + \ldots + p_{m-(n-1)}}{n}$$

Complementary Filter The complementary filter [11] is a frequency domain filter. In its strictest sense, the definition of a complementary filter refers to the use of two or more transfer functions, which are mathematical complements of one another. Thus, if the data from one sensor is operated on by G(s), then the data from the other sensor is operated on by I-G(s), and the sum of the transfer functions is I, the identity matrix. In the case of a one-dimensional filter as will be described in this paper, the identity matrix reduces to the scalar number one. In a typical two-input system, one input will provide information with high frequency noise, and is thus low-pass filtered. The other input provides information with low frequency noise, and is high-pass filtered. If the low-pass and high-pass filters are mathematical complements, then the output of the filter is the complete reconstruction of the variable being sensed, minus the noise associated with the sensors.

3 Implementation

3.1 Notations

In the following, we summarize the notations used in the rest of the thesis.

Definition 1 *Angle of View: In photography, angle of view describes the angular extent of a given scene that is imaged by a camera. It is used interchangeably with the more general term field of view [12].*

Fig. 1. Schematic diagram (Using Taipei 101 as MRP)

Definition 2 *Angle of Object: Angle of object describes the angular extent of a given object that is imaged by a camera. In our work. The angle of object is denoted as θ, which is formed by the lines of the top of the object to the camera and the bottom of the object to the camera. Figure 2 illustrates an example of the angle of object.*

3.2 Measurable Reference Point (MRP)

When we take a photo which contains an image of a known object, it is possible to derive our current position based on the size and angle of the image of the known

object. For the ease of reference, in the rest of this thesis we refer to the known object in the photo as the *Measurable Reference Point* (MRP). Basically, the location, orientation, and the size of the MRP is known in advance for reference. For instance, we know the exact location and the height of Taipei 101 [13]. If our photo contains the image of Taipei 101, we can use it as our MRP to infer our position at the time when the photo is taken. Optical positioning systems that rely entirely on natural features in the images lack robustness, in particular under conditions with varying illumination. In order to increase robustness and improve accuracy of reference points, dedicated MRP are used for systems with demanding requirements for positioning. The MRP serves three purposes for the development of the positioning algorithm: First, it simplifies the automatic detection of corresponding points. Second, the shape, size, and location of MRP have to be known. Third, the MRP can be easily identified and is unique in the positioning environment. One good example of MRP for small and median-scale range is the QR-code, as it is easy to identify and is unique in the target environment.

3.3 Angle Measurement

To calculate the angle of object, we first need to obtain the angle of view. The angle of view can be obtained directly from Android API by calling:

```
android.hardware.Camera.Parameters.getHorizontalViewAngle();
android.hardware.Camera.Parameters.getVerticalViewAngle();
```

Here, we treat the lens as if it were a pinhole at distance S_2 from the image plane. We can obtain S_2 by the following equation.

$$tan(\frac{\theta}{2}) = \frac{d/2}{S_2}$$

Note that once S_2 and the object size d are acquired, the angle of object can be proportionally calculated by the following equation.

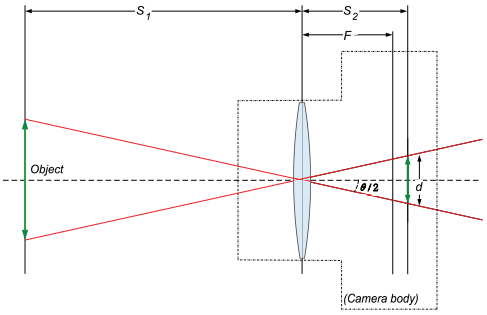

Fig. 2. Angle of object.

$$\theta = 2arctan(\frac{d}{2S_2})$$

3.4 Distance Measurement

In our computation, we use triangle functions to measure the distance between MRP and the camera. Based on the relative orientation of the object to the camera, we separate the distance measurement into three different cases.

Basic Case In this case. the MRP object is facing directly toward the center of the camera. Figure 3 illustrates an example of this case. This is the simplest case for distance measurement. In addition to the size of MRP (denoted as H), the additional required parameter of the camera for visual-based localization includes the view of angle for the object (denoted as θ). We can use the method mentioned in Section 3.3 to calculate θ. Once θ is obtained, the distance between MRP and camera can be computed by the following equation.

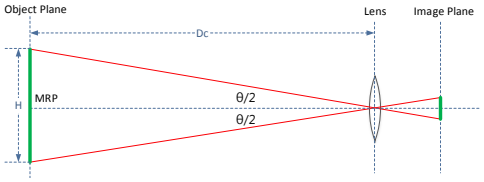

Fig. 3. Measure distance basic case.

$$Dc = \frac{H}{2} / \tan(\frac{\theta}{2})$$

MRP Shifted on Object Plane In this case, the MRP still facing toward the camera, but it is shifted by a distance on the direction perpendicular to the line between the MRP and the camera, as shown in Figure 4. We have found that if a MRP is shifted from the position directly facing the camera to the position up or down by a distance on the direction perpendicular to the line between the MRP and the camera, the projected image will remain the same size.

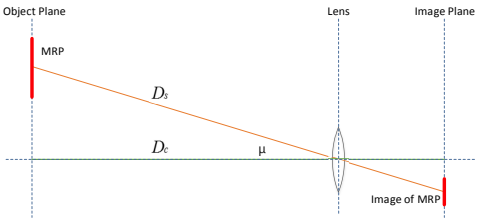

Fig. 4. MRP shifted on object plane.

Theorem 1. *The object has a fixed ratio to its projected image no matter how much of the distance it shifts on the direction perpendicular to the line between the MRP and the camera.*

Proof. Let the object plane be parallel to the image plane. We first assume the distance between the object plane and the lens (denoted as D_o) to be x times of the distance between the lens and the image plane (denoted as D_p). In addition, we assume O_α to be originally facing directly to the camera and its projected image to be P_α, as shown in Figure 5. Based on triangle rules, we have

$$P_\alpha = D_p \tan(\alpha),$$

$$O_\alpha = D_o \tan(\alpha) = x(D_p \tan(\alpha)).$$

Consequently,

$$O_\alpha = x(P_\alpha)$$

Now assume that the object is shifted up by a distance. Without loss of generality, let the new location of the object to be O_β and its projected image

to be P_β. In case when O_α and O_β do not overlap each other, as illustrated in Figure 5, we have

$$\begin{aligned} P_\beta &= D_p \tan(\alpha+\beta+\gamma) - D_p \tan(\alpha+\gamma) \\ &= D_p(\tan(\alpha+\beta+\gamma) - \tan(\alpha+\gamma)), \\ O_\beta &= D_o \tan(\alpha+\beta+\gamma) - D_o \tan(\alpha+\gamma) \\ &= xD_p(\tan(\alpha+\beta+\gamma) - \tan(\alpha+\gamma)). \end{aligned}$$

Consequently,

$$O_\beta = x(P_\beta)$$

As a result, we have

$$\frac{O_\alpha}{O_\beta} = \frac{P_\alpha}{P_\beta}.$$

Based on the above proof, we know that the distance between object plane and camera(D_c) can be calculated directly from the size of the MRP projection.

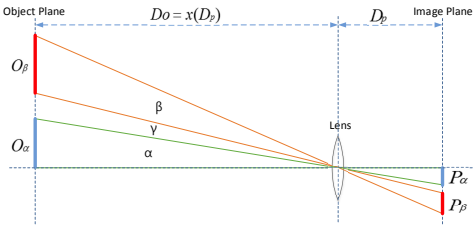

Fig. 5. Proof that object have a fixed ratio relationship with projecting image.

However, the distance between two planes is not the actual distance between MRP and camera. In order to get the real distance from MRP to camera (denoted as Ds), the additional required parameter is the angle between object plane and image plane, denoted as μ. We can obtain μ by detecting the position of MRP in camera image, and then using the angle measurement method mentioned in seciton 3.3. Once μ is obtained, according to the trigonometric functions, Ds can be computed by equation:

$$Ds = \frac{Dc}{\cos(\mu)}$$

Unparalleled Relationship In this case. MRP is not facing toward object plane. We can imagine that it is like MRP is rotated for an angle as shown in Figure 6. In this case, the MRP size we measured L_2 will be smaller than true situation L_1. For the convenience of calculation, we assume that camera lens is located at an infinite distance, then the MRP have an orthogonal projection relationship [14] with the object plane as illustrated in Figure 6. The additional

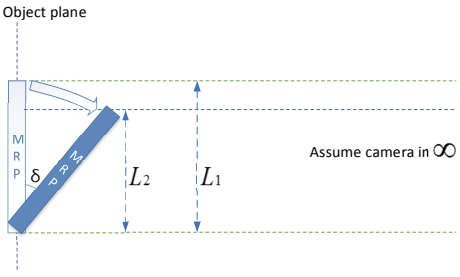

Fig. 6. Measure distance if MRP rotates.

required parameter for this case is the angle between MRP and image plane (denoted as δ).

δ can be obtained by calculating the difference of the orientation between MRP and camera. Unlike the orientation of MRP, which is fixed and known, the orientation of camera is dynamic. We use Android APIs to obtain the most recent orientation of camera. Readers interested in the Android APIs can refer to Android Developer Network [15] for more information regarding Sensors APIs.

Once δ is obtained, according to the trigonometric functions, we can compensate and compute the real MRP image size L_1 using the equation below, then use L_1 to calculate the real distance between MRP and camera:

$$L_1 = \frac{L_2}{\cos(\delta)}$$

3.5 Coordinate System Transformation

Now we have the distance between camera and MRP, the orientation of camera, and the position of MRP in camera image. We can use these data to construct the vector V_1 that extends from the mobile device to MRP. Table 1 summarizes the notation of different positions with respect to various reference systems.

Reference system	Position of MRP	Position of camera
Global reference system	LLA_{MRP}	LLA_{CAMERA}
MRP reference system	$(0, 0, 0)$	(x, y, z)
Mobile reference system spherical coordinate [16]	(D, ζ, η)	$(0, 0, 0)$

Table 1. Comparison of various reference systems

Notice that in the mobile reference system, D denotes the radial distance of MRP from the camera, the ζ and η denote the horizontal and vertical angular extent from the center of image to the position of image that MRP is located (See Figure 7). For orientation angles, u, v, w denote the pitch, roll, azimuth deviation between camera and MRP (See Figure 8).

Note that the global reference position of MRP is known. Hence, if we transform V_1 to the MRP reference system, we can easily convert it to the global reference system.

In the following, we summarize the equations that transform V_1 to the MRP coordinate system:

1. Rotate Y axis (roll angle)

$$x_1 = 0$$
$$y_1 = D$$
$$z_1 = 0$$

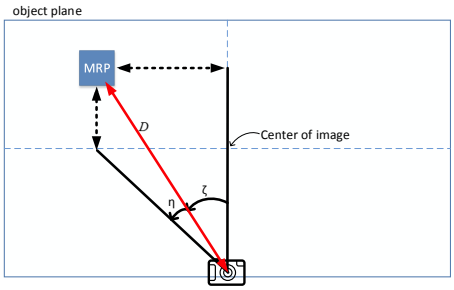

Fig. 7. The mobile reference system.

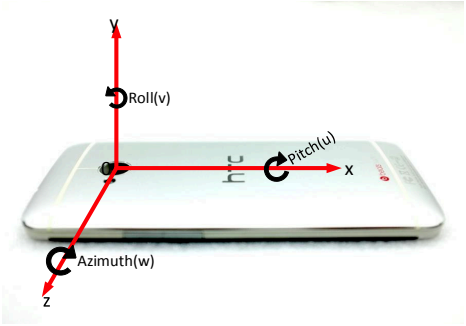

Fig. 8. The orientation of camera. Notice that we use the landscape mode.

2. Rotate X axis (pitch angle)

$$\begin{aligned} x_2 &= x_1 \\ y_2 &= y_1 \times \cos(u - \eta) \\ z_2 &= y_1 \times \sin(u - \eta) \end{aligned}$$

3. Rotate Z axis (azimuth angle)

$$\begin{aligned} x_3 &= -y_2 \times \sin(w - \zeta) \\ y_3 &= y_2 \times \cos(w - \zeta) \\ z_3 &= z_2 \end{aligned}$$

4. Change values order compatible to global reference system Taiwan area(northern/eastern Hemisphere).

$$\begin{aligned} x &= -x_3 \\ y &= -y_3 \\ z &= z_3 \end{aligned}$$

3.6 Hardware

Our hardware environment is the HTC One Android smartphone, which has a build-in camera, gyroscope, geomagnetic field sensors and accelerometers. Table 2 shows the specification of key hardware components that we used in this project. Of course, our software can be deployed not only to the HTC One but also to any other Android devices.

3.7 Software

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. A developer survey conducted in May 2013 found that Android is the most popular platform for

Name	Specification
Camera	Max Picture Size: 2688 x 1520 Max Video Size: 1920 x 1088 Horizontal View Angle: 69.6 Vertical View Angle: 43
Magnetic field sensor	Vendor: Asahi Kasei Microdevices Model: AK8963 3-axis Magnetic field sensor
Accelerometer	Vendor: BOSCH Model: MBA250 3-axis Accelerometer
Gyroscope	Vendor: ST Group Ltd. Model: R3GD20 Gyroscope seosnr

Table 2. Key hardware components used in our visual-based positioning system.

developers, It has been used by 71% of the mobile developer population. Our experimental OS is Android version 4.1.2. The software is developed under The Eclipse IDE with The Android Developer Tools (ADT) plugin [17]. We choose ZXing [18] as our QR-code decoding library. ZXing is an open-source, multi-format 1D/2D barcode image processing library implemented in Java.

3.8 Magnetic Noise Issue

After our early system implementation, we found the issue of the magnetic noise. Magnetic noise is derived from device electronic parts and buildings. Section 2.2 describes the complementary filter which can inhibit magnetic noise.

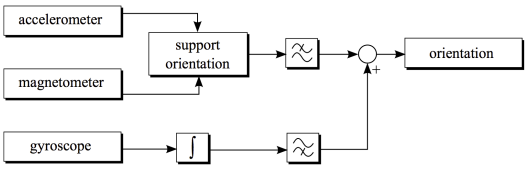

Fig. 9. An overview of the complementary filter.

Complementary Filter for sensor fusion was proposed in [19]. The gyroscope in the device is far more accurate and has a very short response time. Its downside is the dreaded gyro drift. The gyro provides the angular rotation speeds for all three axes. To get the actual orientation, those speed values need to be integrated over time. This is done by multiplying the angular speeds with the time interval between the last and the current sensor output. This yields a rotation increment. The sum of all rotation increments yields the absolute orientation of the device. During this process small errors are introduced in each iteration. These small errors add up over time resulting in a constant slow rotation of the calculated orientation, called the gyro drift.

To avoid both gyro drift and noisy orientation, the gyroscope output is applied only for orientation changes in short time intervals, while the magnetometer/acceletometer data is used as the supporting information over long periods of time. This is equivalent to low-pass filtering of the accelerometer and mag-

netometer signals and high-pass filtering of the gyroscope signals. The overall sensor fusion and filtering are shown in Figure 9.

The sensors provide their data at (more or less) regular time intervals. Their values can be shown as signals in a graph with time as the x-axis, similar to an audio signal. The low-pass filtering of the noisy accelerometer/magnetometer signal (support orientation in figure 9) are orientation angles averaged over time within a constant time window.

Later in the implementation, the low-pass filtering is accomplished by slowly introducing new values from the accelerometer/magnetometer to the absolute orientation:

```
// low-pass filtering: every time a new sensor value is available
// it is weighted with a factor and added to the absolute orientation
accMagOrientation = (1 - factor) * accMagOrientation + factor * newAccMagValue;
```

The high-pass filtering of the integrated gyroscope data is done by replacing the filtered high-frequency component from accMagOrientation with the corresponding gyroscope orientation values:

```
fusedOrientation =
    (1 - factor) * newGyroValue    // high-frequency component
     + factor * newAccMagValue;    // low-frequency component
```

Standard Complementary Filter is effective in electronic parts noise, but it is ineffective for inhibiting magnetic fields changes from buildings with reinforced concrete and steel structures. To solve magnetic fields changes, we prolong the complementary filter computing time intervals to grant the high-frequency values from gyroscope a higher impact. This change has a non-negligible effect in our experiments.

3.9 Gyroscope Offset Issue

One of the downsides of prolonging complementary filter computing time interval is that the effect of gyroscope offset becomes more significant. In order to alleviate the gyroscope offset problem, we sample offset values for a period of time and use their average value for compensation. This simple calibration method can reduce gyroscope offset from 2 seconds per degree to 45 seconds per degree.

3.10 Startup Gyroscope Initial Values

Another downside of prolonging complementary filter computing time interval is that the impact from standard e-compass (accelerometer and magnetometer) becomes minor. The gyroscope initial values should reduce the error from standard e-compass, so we use simple moving average filter (refer to Section 2.2) to obtain better compass results for the startup gyroscope initial values.

4 Performance

To evaluate the performance of the proposed visual-based positioning system, we conducted small, median, and large-scale experiments. The detail setting and results of these experiments are provided in the following subsections.

4.1 Small-scale field experiments

For small-scale field experiments, we choose an 8m x 8m playground as the experiment environment. The MRP is a QR code on a paper of the size 17.65cm x 17.65cm, which is illustrated in Figure 10. The collected data includes standard e-compass (accelerometer and magnetometer) and complementary filter (fused with gyroscope) results. The experiment results are shown in Figure 11 and the statistic results are presented in Table 3.

The results of the small-scale field experiments demonstrate that the error rate

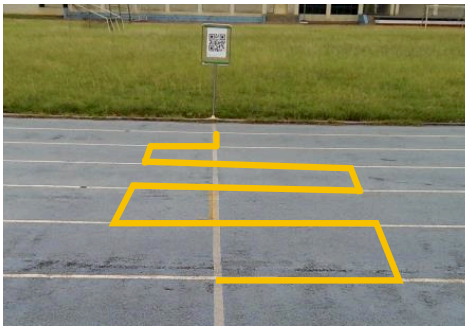

Fig. 10. Small-scale field experiment scenes.

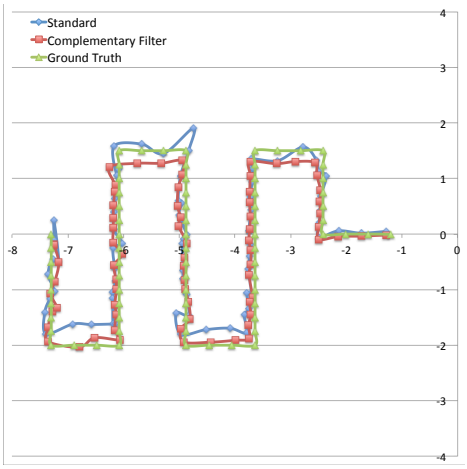

Fig. 11. Small-scale field experiment log chart.

of the positioning is lower than 5%, and the maximal error distance is smaller than 0.5 meters. This results indicate that the system has a high precision for positioning in situations such as in a classroom, a conference room, or a living room.

4.2 Median-scale field experiments

For median-scale field experiments, we choose a 16m x 35m plaza as the experiment environment. The MRP is a QR code on a poster of the size 54.3cm x

Description	Standard	Complementary Filter
Max Error Distance	0.496 meters	0.374 meters
Min Error Distance	0.112 meters	0.022 meters
Average Error Ratio	4.11%	3.61%
Standard Deviation	0.1 meters	0.07 meters

Table 3. Small-scale field experiment statistic results.

54.2cm, as can be seen in Figure 12. Similar to the small-scale experiments, the collected data includes standard e-compass and complementary filter results. In addition, we also collect the GPS results obtained from a *Garmin Dakota 20* navigation device. The experiment and the statistic results are shown in Figure 13 and Table 4, respectively.

The results of the median-scale field experiments demonstrate that the system

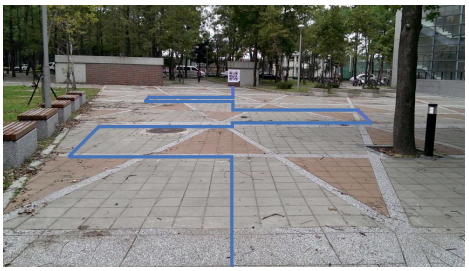

Fig. 12. Median-scale field experiment scenes.

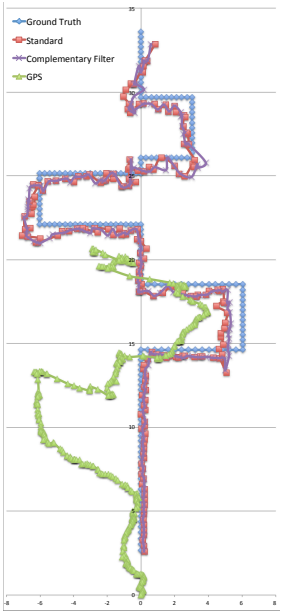

Fig. 13. Median-scale field experiment results.

has a much higher accuracy rate for positioning than the conventional GPS. For instance, the maximal error distance of our system is shorter than 2 meters, which is smaller than even the error of the assisted GPS. This results indicate

Description	Standard	Complementary Filter
Max Error Distance	1.91 meters	1.85 meters
Min Error Distance	0.07 meters	0.09 meters
Average Error Ratio	4.49%	4.18%
Standard Deviation	0.4 meters	0.36 meters

Table 4. Median-scale field experiment statistic results.

that our system are suitable for indoor navigation in a shopping mall or a large exhibit hall.

4.3 Large-scale field experiments

For large-scale field experiments, we choose the NCU Main Library Building as the MRP. We stand on the rooftop of the Engineering Building 5 facing the MRP to obtain our position, as illustrated in Figure 14. The range between the MRP and us is around 800 meters. The experiment results are shown in

Fig. 14. Large-scale field experiment scenes in National Central University.

Figure 15 and Figure 16. In the figures, the main library is denoted by an icon and our true position (obtained by GPS) is denoted by a red flag. Each blue circle indicates a result of the standard method, and each red circle is a result of the complementary filter method. The statistic results are shown in Table 5 and Table 6.

The results of the large-scale field experiments indicate that although the percentage of the error of our system is low (less than 5% in most cases), the absolute error of our system can be quite large when the MRP is far away from the camera. For instance, the maximal error distance is about 40 meters in our large-scale experiments, even when the complementary filter is applied. For applications such as virtual augmentations and location-based services, this

Standard		Complementary Filter	
Error Distance	Error Ratio	Error Distance	Error Ratio
55.94779483	7.63%	41.01243255	5.59%
23.760599	3.24%	30.46624108	4.15%
55.33866059	7.53%	32.29396614	4.40%
16.77831869	2.29%	30.22630963	4.12%
30.18733363	4.11%	29.07574148	3.96%
18.36331176	2.50%	13.21252122	1.80%
22.71171715	3.10%	17.99106596	2.45%
17.22804678	2.35%	14.44225332	1.97%
29.55978553	4.03%	14.68165469	2.00%

Table 5. Large-scale field experiment error ratio.

Description	Standard	Complementary Filter
Max Error Distance	55.95 meters	41.01 meters
Min Error Distance	16.78 meters	13.21 meters
Average Error Ratio	4.09%	3.3%
Standard Deviation	15.33 meters	993 meters

Table 6. Large-scale field experiment statistic results.

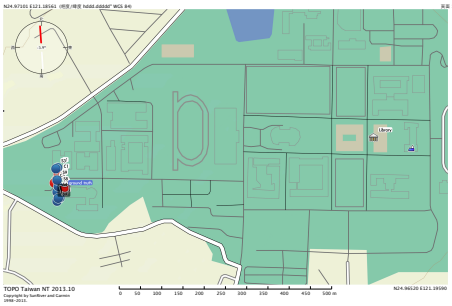

Fig. 15. Campus map to large-scale field experiment results.

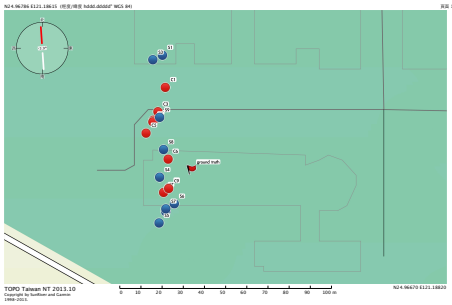

Fig. 16. Zoom in Engineering Building 5 map to large-scale field experiment results.

accuracy is acceptable. However, for applications that require a high precision, our system may not be suitable.

5 Conclusions

In this work, we implement a low-cost visual-based positioning system on an Android-based smartphone. The experiment results show that our system is both accurate and stable for small and median-scale experiments, and has relatively

low error rates in large-scale experiments. The experiment results support that our system is suitable for applications such as indoor navigation in a shopping mall or an exhibit hall, and is accurate enough for applications such as outdoor virtual augmentations and location-based services. In the future, we would like to improve our system to provide even more stable and accurate positioning. For example, multiple MRPs may be used to obtain a better accuracy. Another possible approach is to integrate our system with map matching [20] for a higher positioning precision. In addition, we would like to come up with a solution to better eliminate the magnetic noise or hybrid our system with the dead reckoning algorithm [21]. At current stage, our system is more of a proof-of-concept. It does not have a backend database with a large number of MRP entries. To make our system really useful, we would like to incorporate a MRP database and the pattern matching capability to our system in the future. Our source code is already public and published [22].

References

[1] Ferris, B., Fox, D., Lawrence, N.D.: Wifi-slam using gaussian process latent variable models. In: IJCAI. Volume 7. (2007) 2480–2485

[2] Bekkelien, A., Deriaz, M., Marchand-Maillet, S.: Bluetooth indoor positioning. (2012)

[3] Ganick, A., Ryan, D.: Light positioning system using digital pulse recognition (January 31 2013) US Patent 20,130,028,475.

[4] Racoma, J.A.: Indooratlas uses magnetic fields for location-awareness in buildings and large structures (2013)

[5] Mulloni, A., Wagner, D., Barakonyi, I., Schmalstieg, D.: Indoor positioning and navigation with camera phones. Pervasive Computing, IEEE **8**(2) (2009) 22–31

[6] Kawaji, H., Hatada, K., Yamasaki, T., Aizawa, K.: An image-based indoor positioning for digital museum applications. In: Virtual Systems and Multimedia (VSMM), 2010 16th International Conference on, IEEE (2010) 105–111

[7] Bilton, N.: Behind the google goggles, virtual reality. Published: February **22** (2012)

[8] Chen, D.M., Baatz, G., Koser, K., Tsai, S.S., Vedantham, R., Pylvanainen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., et al.: City-scale landmark identification on mobile devices. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 737–744

[9] Hile, H., Borriello, G.: Positioning and orientation in indoor environments using camera phones. Computer Graphics and Applications, IEEE **28**(4) (2008) 32–39

[10] wikipedia: Simple moving average

[11] Glasser, P.C.: An introduction to the use of complementary an introduction to the use of complementary filters for fusion of sensor data. http://www.glassercommunications.com/paul/samples.htm

[12] wikipedia: Angle of view

[13] wikipedia: Taipei 101

[14] wikipedia: Orthographic projection

[15] Developers, A.: Sensors overview — android developers. `http://developer.android.com/guide/topics/sensors/sensors_overview.html` (2013)

[16] wikipedia: Spherical coordinate system

[17] Developers, A.: Android developer tools (adt) plugin for eclipse. URl: http://developer.android.com/tools/ (2013)
[18] Mackintosh, A., Martin, A., Brown, B., Brunschen, C., Daniel, S.: Zxing, open source library to read 1d/2d barcodes (2012)
[19] Colton, S., Mentor, F.: The balance filter. Presentation, Massachusetts Institute of Technology (2007)
[20] Bernstein, D., Kornhauser, A.: An introduction to map matching for personal navigation assistants. (1998)
[21] Jin, Y., Toh, H.S., Soh, W.S., Wong, W.C.: A robust dead-reckoning pedestrian tracking system with low cost sensors. In: Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on, IEEE (2011) 222–230
[22] Chien, C.S.: A visual-based positioning system source code. https://github.com/wasn-lab/ (2013)