ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.19, 1–8

Vision-based refinement of GPS location and compass orientation

Jun Park[a)]

Computer Engineering Department, Hongik University,
94 Wow-san-ro, Mapo-gu, Seoul, 121–791, Korea
a) *jpark@hongik.ac.kr*

Abstract: Most commercially available Global Positioning Systems (GPS) do not provide location information that is sufficiently accurate to be used for practical location based services (LBS). Especially when there are high building structures nearby, GPS location measurements are known to be erroneous. In this paper, we present a computer vision based method for refining user's two dimensional location and one-dimensional orientation starting from inaccurate GPS and digital compass measurements. Our method utilizes corner positions of buildings in the digital map and the building vertical edges in the captured images. Once calculated, refined user's position and orientation can be used as an initial value for sensor-based tracking in accordance with user's panning motion.

Keywords: GPS, digital compass, localization, location based service
Classification: Electron devices, circuits, and systems

References

[1] Z. Hu and K. Uchimura: Int. J. ITS Res. **4** (2006) 3.
[2] G. Reitmayr and T. Drummond: IEEE ISMAR (2006) 109. DOI:10.1109/ISMAR.2006.297801
[3] J. Karlekar, S. Zhou, W. Lu, Z. Loh and Y. Nakayama: IEEE ISMAR (2010) 175. DOI:10.1109/ISMAR.2010.5643567
[4] A. J. Davison, I. D. Reid, N. Molton and O. Stasse: IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1052. DOI:10.1109/TPAMI.2007.1049
[5] R. O. Castle, G. Klein and D. W. Murray: IEEE ISWC (2008) 15. DOI:10.1109/ISWC.2008.4911577
[6] T. Langlotz, C. Degendorfer, A. Mulloni, G. Schall, G. Reitmayr and D. Schmalstieg: Comput. Graph. **35** (2011) 831. DOI:10.1016/j.cag.2011.04.004
[7] C. Arth, M. Klopschitz, G. Reitmayr and D. Schmalstieg: IEEE ISMAR (2011) 37. DOI:10.1109/ISMAR.2011.6092368
[8] A. Brilhault, S. Kammoun, O. Gutierrez, P. Truillet and C. Jouffrais: 4th IFIP International Conference on New Technologies, Mobility and Security (2011) 1. DOI:10.1109/NTMS.2011.5721061
[9] H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Kosecka and G. Borriello: Pervasive Computing **5538** (2009) 59. DOI:10.1007/978-3-642-01516-8_6
[10] A. Millonig and K. Schechtner: IEEE Intelligent Transportation Systems (2005) 43. DOI:10.1109/TITS.2006.889439
[11] P. Perona and J. Malik: IEEE Trans. Pattern Anal. Mach. Intell. **12** (1990) 629. DOI:10.1109/34.56205

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

[12] http://www.nsic.go.kr/.
[13] A. Guttman: Proc. of the 1984 ACM SIGMOD International Conference on Management of Data **14** [2] (1984) 47. DOI:10.1145/602259.602266

1 Introduction

With rapid prevalence of smartphones, Location Based Service (LBS) became one of the most promising applications where users can benefit from up-to-date point of interest (POI) and navigation information overlaid on their smart phone camera views. However, Global Positioning System (GPS) position estimates are known to be erroneous especially in urban canyon environments where high buildings occur in close proximity. The errors are often greater than 20 m, preventing meaningful location based services.

An alternative or supporting approach is vision-based. Street lanes can be used as references for vehicle navigation applications [1]. However, street lanes are not often visible from pedestrian's viewpoints. When 3D models of buildings are available, model-based methods may be used for robust 6DoF tracking [2, 3]. However, 3D models are rarely available for most outdoor environments. Simultaneous localization and mapping (SLAM) technologies can be used for tracking location and building environment maps simultaneously [4]. Our approach differs from SLAM for utilizing existing digital maps. SLAM based approaches may be also used for tracking purpose based on pre-created maps. For example, Castle et al. enabled selection of the correct local map from several environment maps composed of thousands of point features [5]. Although this technique provides accurate tracking information, map building is a labor-intensive task and point maps are not available in most environments. Panorama image based tracking is also feasible without any known 3D feature locations [6, 7]. However, the revisited user location needs to be very close to the original location where the panorama image was created. Landmarks could be also used to enhance positioning accuracy [8, 9]. Brilhault et al. developed a blind pedestrian positioning system that utilizes GPS receiver and wearable sensors (accelerometers, electronic compass and pedometer) combined with landmark detection and a map matching method [8]. Preliminary results showed that 80% of the positioning errors were less than 0.5 m. Although landmarks may provide accurate pose, their locations need to be calibrated in advance to provide pose information. A system for measuring the quality of the landmarks may be also required to avoid ambiguous landmarks misleading the users [10].

In summary, 3D models, 3D point maps, panorama images, and landmarks are not generally available for urban environments. However, digital maps that include building corner GPS positions are often available. In this paper, we present a computer vision based method for refining user's two-dimensional location and one-dimensional orientation starting from inaccurate GPS and digital compass measurements. Our method is based on pairs of a building edge position in the image and its corresponding GPS position. As long as three or more of these pairs are available, erroneous GPS position and compass orientation can be largely

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

corrected. Refined position and orientation can be used as initial values for sensor-based LBS tracking: sensor-based tracking is practical because users tend to stay at one location while taking panning motions. Through panning motions, users could observe POI information in the areas where building edges are not visible.

2 Building edge detection

2.1 Noise filtering

In general, building detection in urban environments is not trivial. More than anything else, images taken from mobile devices are often involved with noise. Simply smoothing out the whole images will also degrade edges. We applied anisotropic diffusion [11], which reduces image noise while preserving edges. Anisotropic diffusion is defined as the following equation.

$$\frac{\partial}{\partial t} I(\bar{x}, t) = \nabla \bullet (c(\bar{x}, t) \nabla I(\bar{x}, t)) \tag{1}$$

Here, $I(\bar{x}, t)$ is the image, $\bar{x}$ is the pixel position in the image, and t represents iteration step. $c(\bar{x}, t)$ is a diffusion function, which monotonically decreases with absolute gradient values of the image. This function may adjust the degree of diffusion based on the image gradient. We used the following diffusion function.

$$c(\bar{x}, t) = \frac{1}{1 + \left(\frac{|\nabla I(\bar{x}, t)|}{\kappa} \right)^{1+\alpha}} \tag{2}$$

Diffusion effect is large when image gradient is similar to diffusion constant κ ($|\nabla I| \approx \kappa$), hence noise is reduced and edges are preserved if κ is similar to the gradient of the noise. In our experiments, diffusion constant κ was empirically determined.

2.2 Vertical edge detection

After removing noise, vertical edges were detected by applying edge filters: Canny edge detection in grey-scale images. Edges were categorized into horizontal and vertical edges to be used for building vertical edge extraction. After edges were detected, we applied a mask to remove short, adjacent edges, and then applied Hough transform to obtain line equations.

Horizontal edges were utilized for two purposes: to detect building corner edges inside building areas; to reject incorrectly detected vertical edges. Unlike edges adjacent to sky area, building corner edges within building areas are hardly detected because two building sides are often in uniform colors. Because horizontal edges detected from building windows from two building sides are not parallel, connecting these line intersections forms a vertical line, which is the building corner edge inside the building region (Fig. 1 – edge #7).

Horizontal edges may be also used for rejecting vertical edges that are not building corner edges. If a horizontal edge crosses any vertical edges, those vertical edges were considered as non-corner edges and rejected (Fig. 1 – edge #2 and #4).

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

Fig. 1. Utilization of horizontal edges: horizontal edges were used for detecting building corner edges inside building region (edge #7) and rejecting non-corner edges (edge #2 and #4)

3 Correspondence (grid matching)

We assumed that the candidates of the true user position are those within the circular error boundary from the measured GPS position. We chose candidate positions for each 5×10^{-5} GPS position in longitude and latitude (approximately 5.5 m in our environments). Digital maps were used to extract visible building corners at each candidate position. We obtained digital maps from Korean National Spatial Information Cleaninghouse (NSIC) [12].

For real-time implementation, we have pre-constructed visible corner maps that contain angular directions (for every 1° in 360° orientation) and distances to visible building corners from the candidate positions. We have chosen 1° step in order to ensure 1.0 m precision for building corner positions with 30 m GPS errors (1° corresponds to 0.52 m in 30 m). However, the gaps between most buildings were larger than 2 m. System performance was not affected much with 2° step in our experiments. Angular steps may be adjusted according to the minimum distance between building corners in the environment.

By updating the minimum distance from a given position in each direction, we could maintain the closest visible building corners. While updating a visible corner map, building faces that connects two corners were maintained also. That is, if a building corner was behind a face, that corner was considered to be occluded by this front building. In this research, we ignored building heights because the current version of the digital map did not have information on building heights.

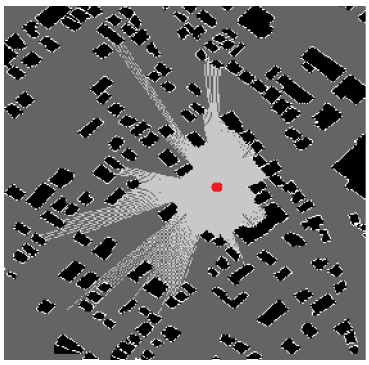


Fig. 2. An example of visible corner map: minimum distances to surrounding buildings from a candidate position (depicted as a red dot) are indicated in bright grey lines

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

Fig. 2 shows an example of a visualized visible corner map. From the red dot position, distances to the closest buildings were indicated in bright grey lines. In the image, the dark grey background is road area, and black polygons are buildings.

Pre-constructed visible corner maps were used to select positions whose visible building corners match with the extracted building edges. Visible corner maps for each position contain information on visible building corner angular positions relative to the North. Based on angular distance differences (RMS errors), matching tests were performed between detected vertical edges and building corners in the map. If the following equation is satisfied, then matching is successful.

$$\left(\sum_{k=1}^{n} \sqrt{\left(\theta_k - \arg\min_{\varphi_j \in \Psi} |\theta_k - \varphi_j| \right)^2} \Big/ n \right) < \tau \tag{3}$$

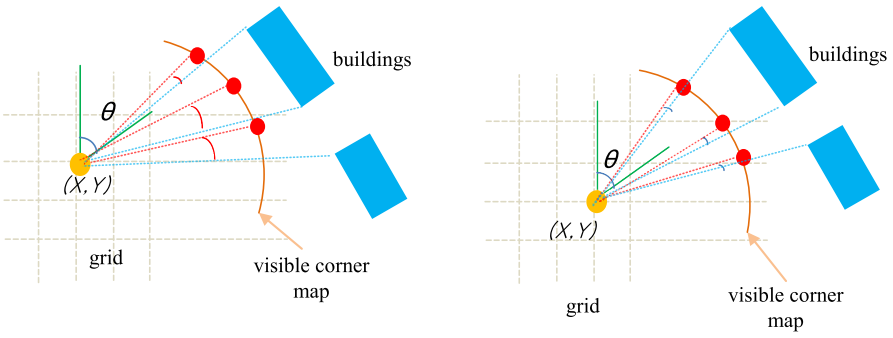
Here, θ_k is the angle for k^{th} edge detected from the image, φ_j is an angle from the visible corner map Ψ, and τ is the threshold value. We maintained a matched corner list from the visible corner map so that no more than one edge matches to a single corner. Fig. 3(a) and (b) show two matching cases: matching failure and success.

Only a few (mostly less than ten) positions matched successfully: when 3 edges were detected, 6 to 12 positions matched successfully; when 4 edges were detected, less than 4 positions matched successfully in our experiments. The matched solutions were passed onto the optimization phase (explained in the next section).

GPS error boundary and orientation error range were obtained through experiments.

Sizes of the pre-computed visible corner maps for $60 \times 60\,m^2$ area was between 3 KB to 10 KB depending on the number of buildings in the area. Computation time for constructing $60 \times 60\,m^2$ area visible corner map was about 29 seconds on a mobile device (Samsung Galaxy Note 10.1).

To test the feasibility of online depth map computation, we previously applied R-tree technique (one of the most popular access methods for rectangles [13]). Given a viewing direction, R-tree reduced the search range of visible buildings, and hence computation time was decreased from 29 to 3.5 seconds. However, computation time was still too long for real-time implementation. Considering that the size



(a) Matching failure (b) Matching success

Fig. 3. Correspondence
(a) Matching failure where RMS > τ
The angular RMS error is larger than the threshold
(b) Matching success where RMS < τ
The angular RMS error is smaller than the threshold

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

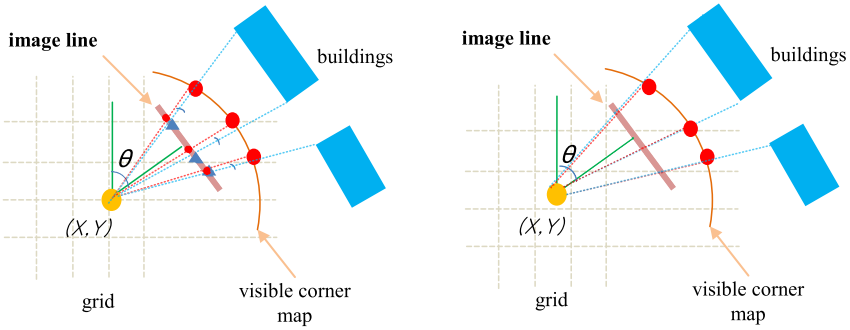
of the pre-computed visible corner maps for $60 \times 60\,\mathrm{m}^2$ area was mostly less than 10 KB, we determined that off-line, pre-computation of depth maps was more practical. Once visible corner maps are downloaded from the server, users may look around in 360° and in a few step distances.

4 Optimization

Successfully matched positions were used for convergence to optimal solutions. For tracking 3DoF pose (instead of 6DoF) and using vertical edges only, we used 1D image space (an image line, see Fig. 4-a). Image line is a 2D analogy of image plane in 3D space.

We assumed that the camera focal length is fixed and known in advance. Based on the matched position and orientation (X, Y, θ) and the camera focal length (f), the equation of the image line can be obtained (a line that passes through $(X + f\cos\theta, Y + f\sin\theta)$, and whose slope is $-\cot\theta$ in 2D space).

Rays were formed from the focal point to the building corner angular positions in the visible corner map $(r_i, i = 1 \ldots n \quad n \geq 3)$. Then we calculated the intersections (red dots in Fig. 4-a) of these rays with the image line (denoting the intersections as (t_i), $i = 1 \ldots n$). These intersections perfectly match with the building edge's horizontal coordinates $((u_i)$, $i = 1 \ldots n$, (blue triangles in Fig. 4-a)) if the position and orientation are accurate: if the position is the true position, blue triangles and red dots should coincide on the image line. The differences $((t_i - u_i)$, $i = 1 \ldots n)$ were used for finding accurate position and orientation by applying Levenberg-Marquardt nonlinear optimization. Fig. 4-b shows an optimized result where the refined position was moved from the grid position of Fig. 4-a.



(a) After matching and before optimization (b) After optimization

Fig. 4. Optimization
(a) After matching: same as Fig. 3(b) – the position is on the grid
Building corner positions (red dots) and detected edge positions (blue triangles) are marked on the image line
(b) After optimization: improved accuracy
The refined position was moved from the grid position of (a)

5 Experiments and results

We performed experiments in downtown areas where tall buildings (higher than 30 m) are located. We used 2σ GPS error boundary with estimated standard

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

deviation which was obtained by collecting GPS measurements at a single location in a long period of time (e.g., in one experiment, we used $\sigma = 14.8$ m). A pre-computed visible corner map was used to obtain visible building corner angular positions from each user candidate positions within the GPS error boundary. For experiments, a camera and a GPS receiver of Samsung Galaxy Note 10.1 were used. We tested our system at three locations using a panning arc of roughly 30°. We repeated four times in each situation collecting 100 to 200 measurements per test.

Fig. 5-left shows an example of 3DoF pose improvement result. POI texts in red and yellow represent before and after pose refinement respectively (in this experiment, FedEX was the POI). Fig. 5-right shows another example at a different location. Four edges were detected producing a corrected POI display (corrected from cyan text position to orange text position).

Testing our system at different locations and in different directions, about 78% of the GPS measurements were converged closer to the correct positions. We measured the improved distances and angles instead of the absolute errors because it was difficult to measure the ground true positions and orientations precisely. On average, our method improved 13.7 m in position and 0.91° in orientation from GPS and compass measurements. Averages and standard deviations of the improvements are shown in Table I.

Orientation improvement was small because digital compass measurements were fairly accurate. Standard deviation of the orientation improvement was larger than 1.0° because refinement was often larger than 2°.

Fig. 5. Examples of 3DoF pose improvement results: pose improvement based on three vertical edges (left) and four vertical edges (right)

Table I. Refinement results (improved from raw measurements)

	Mean	Standard Deviation
Distance	13.7 m	4.7 m
Angles	0.91°	1.1°

6 Conclusion and discussion

In this paper, we introduced a 3DoF pose (2D position + 1D orientation) computation method for LBS in urban canyon environments. Our proposed method is

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014

based on erroneous position and orientation estimates obtained from a GPS receiver and a digital compass, and building vertical edges extracted from an average field-of-view camera.

Around 78% of the GPS measurements were correctly improved when three or more building edges were detected. On average, our method improved 13.7 m in position and 0.91° in orientation from GPS and compass measurements.

Our method can be used for providing accurate initial position and orientation for sensor-based LBS tracking. Through panning motions, users may observe POI information in the areas where building edges are not visible.

Other methods that are based on 3D models, panorama images, or landmarks may also be used to enhance positioning errors. However, it is necessary to prepare the environments in advance such as constructing 3D models, taking panorama images, or calibrating landmark positions/orientations. One of the advantages of our method is that it is based on commercial-off-the-shelf digital maps.

There are some limitations to the presented method. Firstly, in some situations, GPS position errors were larger than 30 m or even larger than 50 m. In these cases, true user positions were outside the GPS error boundary. Expanding GPS error boundary resulted in too many candidate positions and sometimes local-minima convergence. Secondly, in some urban canyon environments, buildings were positioned too close to one another. In such cases, some edges were undetected and others were incorrectly interpreted. As a future work, we plan to determine tight GPS error boundaries in order to reduce the number of candidate positions and enhance system performance.

Acknowledgments

This work was supported by 2013 Hongik University Research Fund.

DOI: 10.1587/elex.11.20140566
Received June 13, 2014
Accepted September 1, 2014
Publicized September 18, 2014
Copyedited October 11, 2014