15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Pedestrian Positioning in Urban City with the Aid of Google Maps Street View

Haitao Wang
The University of Tokyo
Komaba, Meguro-ku, Tokyo
haitao.wang@kmj.iis.u-tokyo.ac.jp

Yanlei Gu
The University of Tokyo
Komaba, Meguro-ku, Tokyo
guyanlei@kmj.iis.u-tokyo.ac.jp

Shunsuke Kamijo
The University of Tokyo
Komaba, Meguro-ku, Tokyo
kamijo@iis.u-tokyo.ac.jp

Abstract

Pedestrian navigation has become one of the most used services in people's city lives. Not only smartphone based navigation, but also the application in the next generation of intelligent wearable devices, such as smart glasses, attract attentions from both scientists and engineers. The satisfied navigation service requires an accurate positioning technology. Even though the current smartphones have integrated various sensors, such as Global Navigation Satellite System receiver, gyroscope, accelerometer and magnetometer sensors, the performance of positioning in city urban is still not satisfied. The reasons of the errors include GNSS signals reflections, high dynamic of pedestrian activities and disturbance of the magnetic field in city environments. This paper proposes to utilize the camera sensor for improving the accuracy of the positioning. The camera sensor provides the visual observation for surround environment. This observation is compared with the available Google Maps Street View in order to correct positioning errors. With the visual matching between the geo-tagged pedestrian's photo and the reference images from Google Maps Street View, we expect to reduce the positioning error into 4 meters, and further recognize which side of the road or which corner of the crossroads the pedestrian is in.

1. Introduction

Nowadays, the pedestrian navigation application becomes one of the most used and convenient service in people's city lives. The quality of navigation service significantly depends on the accuracy of pedestrian localization. Global Navigation Satellite System (GNSS) is the most widely applied positioning system in the smart devices such as smartphone and smart glasses. GNSS can perform high accuracy with a standard deviation about 0.3 meter in the open sky condition. However, in the urban city, GNSS's signals are usually influenced by the blockage and the reflection of high buildings. The huge error of GNSS based positioning will lead to a mistake in distinguishing the correct side of road and intersection [1]. There are some other commercialized positioning technologies, including WiFi and Pedestrian Dead Reckoning (PDR) [1, 2]. They also do not have satisfied performance in the urban areas. To solve this problem, this paper proposes a pedestrian positioning system with the technique of visual localization, using the matching between the geo-tagged photo from the smartphone and the reference images from Google Maps Street View.

Google Maps Street View is a comprehensive and large database provided by Google, which consists of geo-tagged 360° panoramic images of almost all main streets and roads in a number of countries. The panoramic images of Google Maps Street View are recorded by a spherical arrangement of cameras and the localization comes from the high-performance localization system. Because of its characteristics and quality, Google Maps Street View attracts more and more attentions in the field of computer vision and localization.

Majdik et al. used Street View images to localize a Micro Aerial Vehicle by matching images to Google Maps Street View images and added 3D models of buildings as input to improve its accuracy [3]. Torii et al. matched descriptors computed directly on queried image and multiple Google Maps Street View panoramas with learning a distinctive bag-of-word model to localize [4]. However, these methods only solved the place recognition problem and provided topological localization via image matching.

Other researches focus on localizing images in large scale metrical maps built from Structure from Motion. Irschara et al. built accurate point clouds using structure from motion and then computed the camera coordinates of the query image [5]. Zamir et al. constructed a dense map from 100,000 Google Maps Street View images [6]. They provided a reliability score of the results based on the voting distribution function and finally determined the best result with highest number of votes. Agarwal et al. estimated the 3D position of tracked feature points from short monocular camera sequences and then computed the rigid body transformation to the Google Maps Street View panoramas [7]. Other works with a rough GNSS input localization and outdoor information (2D/3D models) for a precise pedestrian localization are also proposed [8, 9].

Rather than other researches relying on accurate maps built with a large amount of overlapping geo-tagged images, the main contribution of this paper is to estimate the accurate positions on the basis of only a few reference Google Maps Street View images, which are side view pictures. In this paper, we proposed a method to determine the pedestrian position from the image matching between camera and Google Maps Street View. In the proposed method, each geo-tagged query image was recorded by the camera of a smartphone. Then we applied the Affine-SIFT (ASIFT) [10] method to find the matching from the descriptors of query image to the descriptors of Google Maps Street View images. Finally, we could estimate the pedestrian position by decomposing the Homography matrixes. To verify the proposed method, the developed system is tested in an urban environment of Tokyo city.

The remaining part of this paper is organized as follows: Section 2 describes the proposed method, including the parts of reference dataset building, feature based image matching and camera pose estimation. In section 3, the experiments and results are explained. The paper is ended up with conclusion and future work in section 4.

2. Proposed Method

In our method, for each query image recorded by the camera of a smartphone, its geo-tag is determined by the GNSS receiver in the same smartphone. Considering the error of the GNSS positioning result, we download multiple reference images near their geo-tags from Google Maps Street View to match with them and estimate camera pose from the best matching results, expecting to improve the positioning accuracy.

2.1. Dataset from Google Map Street View

To correct the positioning from image matching result, the reference dataset should contain the geographical coordinates and heading information of images. Google Maps Street View provides intermittent panoramic images with a distance from 5 to 20 meters. These panoramic images can be downloaded as normal images with the use of Google Street View Image API [11] by inputting the parameters of image size, latitude, longitude, field of view, heading angle and pitch angle. We can also determine the direction of each road with the help of Google Map. For each panoramic image, we download two images with the heading perpendicular to the road as the dataset. It means our dataset is composed of images of building's walls next to the road. Then, each reference image is tagged with its latitude, longitude and heading angle. As for the field of view and pitch angle, in our experiment, reference images with 90 degree of field of view and 10 degree of pitch angle are used.

When pedestrian walks, query images and positioning results are synchronously collected. The images with position become the geo-tagged query images. For positioning each image, we determine the interested roads with vertical distance less than 20 meters from positioning result, as shown in Figure 1. Then matching algorithm compares the query image with the reference images from Google Map Street View with the distance less than 10 meters to the vertical point.

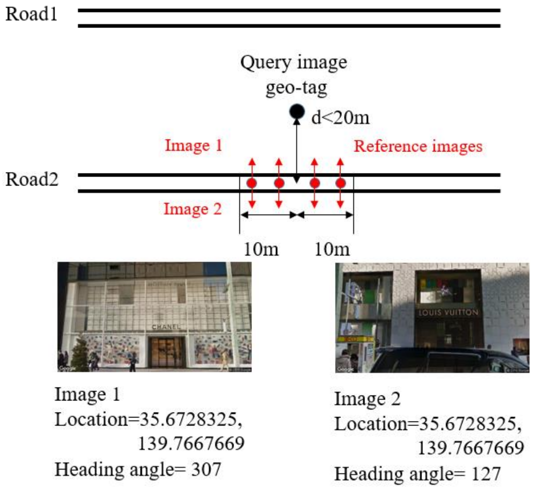

Figure 1. Illustration of the generation of the dataset from Google Map Street View.

2.2. Feature Based Image Matching

After building the reference dataset for the input query image, the next step is to do image matching and find a best match. There are many approaches for image matching and generally they have two main steps: key feature extraction and matching. SIFT (Scale-Invariant Feature Transform) features is widely used [4, 5] and some of researches [3, 6, 7] choose to use FLANN (Fast Library for Approximate Nearest Neighbors) to find the nearest neighbors for each SIFT descriptors. However, the methods like SIFT or MSER (Maximally Stable External Regions) [12] are faced with a challenge when the view point changes a lot. In this kind of situation, they cannot find enough matched key features.

Considering that the behaviors of pedestrians are complex and we hope to find a match in most of situations, so we choose to use the ASIFT method [9]. ASIFT simulates a set of sample views of the initial images, obtainable by varying the two camera axis orientation parameters, namely the latitude and the longitude angles. Then it applies the SIFT method itself to all these images. ASIFT shows good performance to find key features matches in the situation with a large transition tilts.

Therefore, we applied the ASIFT method to compute SIFT descriptors and find the match from the query image to the reference images. To improve the feature correspondences result, we use RANSAC (Random sample consensus) to filter out the outlier matches. Then, reference images with less than 10 feature matches with the query image will be discard. In most case, only a few reference images can be left. The next step is to use these rest reference images to estimate the camera pose.

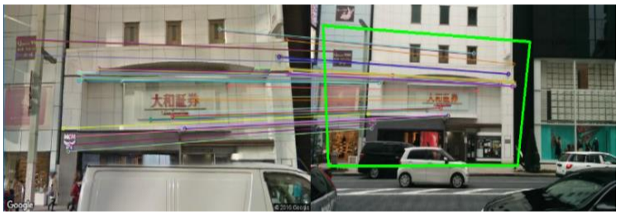

Figure 2. Matching between a reference image (left) with the query image (right). Projection of the reference image on the query image is shown in the green window.

2.3. Camera Pose Estimation

Both the query image recorded by the camera and the downloaded reference images from Google Maps Street View can be considered as taken from a pinhole camera model.

As shown in Figure 3, the feature in the world coordinates $P = (X, Y, Z)$ can be projected into the pixel coordinates (u, v) as

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

where

$$(R|t) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \quad (2)$$

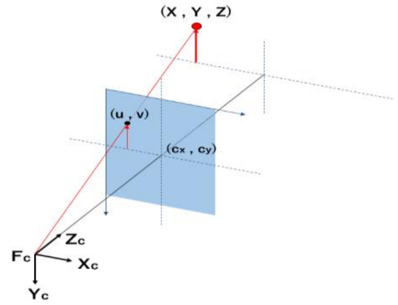

Figure 3. The geometry of a pinhole camera model.

is the rotation-translation matrix, which is used to describe the camera motion around a static scene.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

is the camera intrinsic parameters matrix, where (c_x, c_y) is the principal point in the image and f_x, f_y are the focal lengths expressed in pixel units.

When features in different images are matching, we can consider that these matched features are the same feature $P = (X, Y, Z)$ in the world coordinates. Because the reference dataset is composed of images perpendicular to the building's walls, to simplify the problem, we can assume that all the features are places on a plane, whose $Z = 0$. So we can get features in the query image as (u_c, v_c):

$$\begin{pmatrix} u_c \\ v_c \\ 1 \end{pmatrix} \sim K_c (r_{\mathrm{c},1}, r_{\mathrm{c},2}, t_c) \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{4}$$

where K_c is the camera intrinsic parameters matrix of the user's camera, which is known. $(R|t)$ is the rotation-translation matrix from the camera pose of reference image to the camera pose of query image, which we want to know. And r_1, r_2 are the first and second column of the rotation matrix. Our main goal is to estimate the position of the query image from the translation matrix and this assumption does not affect the values in the translation matrix. So this simplification can work in our research.

In a same way, features in the reference image can also be projected as (u_g, v_g):

$$\begin{pmatrix} u_g \\ v_g \\ 1 \end{pmatrix} \sim K_g (r_{\mathrm{g},1}, r_{\mathrm{g},2}, t_g) \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{5}$$

where K_g is the camera intrinsic parameters matrix of Google Maps Street View. In this research, we use offline calibration to get the focal lengths and select the center of the image as the principal point. The geographic coordinates and heading angle of each reference images which we have known are related to $(r_{\mathrm{g},1}, r_{\mathrm{g},2}, t_g)$.

In the image matching, if multiple features pairs has been matched, the perspective transformation between two images can be calculated as a Homography matrix H. So we can associate (u_g, v_g) and (u_c, v_c) by the 3×3 matrix H as:

$$\begin{pmatrix} u_c \\ v_c \\ 1 \end{pmatrix} \sim H \begin{pmatrix} u_g \\ v_g \\ 1 \end{pmatrix} \tag{6}$$

By jointing formula (4), (5) and (6), we can get the rotation-translation matrix as

$$(r_{\mathrm{c},1}, r_{\mathrm{c},2}, t_c) \sim K_c^{-1} H K_g (r_{\mathrm{g},1}, r_{\mathrm{g},2}, t_g) \tag{7}$$

We could obtain the translation matrix from equation (7). Thus, we can estimate a pair of latitude and longitude from the relative offset between the query image and a matched reference image. After the image matching step, there are only a few reference images from close positions left in most case. Finally, we use the weighted average method to get pedestrian positioning.

3. Experiment

3.1. Experiment Setup

This research selected the Ginza area in Tokyo as the experiment spot. Because of the density of the tall and similar buildings, pedestrians are often confused to find their position and destination in this area. Our goal is to estimate an accurate positioning result in this kind of areas. In addition, there are a lot of shopping malls in this area, which also facilitates the development of our navigation system in the future.. Therefore, we chose Ginza area as the testbed.

In the experiment, the smart phone Google Nexus 5 was used. The image matching was conducted based on images captured from the camera on this smart phone. To simulate the situation when a visitor is walking on the sidewalk in a new environment, we looked around while walking in the trajectory as shown in Figure 4 to capture images from the camera and collect GNSS positioning data simultaneously. The walking distance is about 500 meters. The system time of the smart phone was used for synchronization. In pedestrian positioning and navigation, it is more important to distinguish which side of the road the pedestrian is on. This paper adopts the distance error and correct side rate to evaluate the performance of the proposed method.

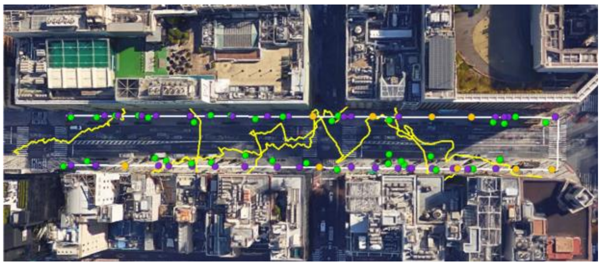

Figure 4. Visualization of results: walking trajectory (white line), GNSS result (yellow line), image matching result (green dot), ground truth (purple dot), and unmatched ground truth (orange dot).

3.2. Evaluation for the Performance of the proposed method

Figure 4 shows the positioning results in the experiment. The white line is the ground truth route and the purple dots are the ground truth positions where we captured images. The yellow line is the GNSS positioning result directly collected by the smart phone, from which we cannot distinguish which side of road we were. Green dots are the positioning result based on this paper.

Table 1 shows the quantitative comparison of our proposed method and the GNSS positioning result. Our proposed method shows extremely good performance in the urban positioning when comparing with GNSS. The positioning mean error is reduced to about 4-meter. In addition, this method can achieve 91% correct side rate when the image matching result exists. From Table 1, we can see that the positioning availability is not 100%, because the images captured from some of the places cannot find a match from the references images. The reason in most of this kind of situations is coming from the quality of the reference. When the camera of Google is too close facing a wall, there will be less interest features to match with the query image. In addition, if either the query image or the reference image is in a strong sunlight or a deep shadow, the image matching will also be influenced.

We also evaluated our method with query images from different view directions. As shown in Table 2, when the camera direct to the side view building (view angle=0°), the positioning achieved the best performance. This conclusion can suggest the development of street view matching based localization method.

Table 1. Evaluation of the proposed method.

	Positioning Availability	Correct Side Rate	Mean Error (Meter)	Standard Deviation (Meter)
Image Matching	81.8%	91.7%	4.37	2.49
GNSS	100%	8.3%	26.72	8.99

Table 2. Evaluation of the proposed method with different view directions from side view.

View angles from side view	Positioning Availability	Correct Side Rate	Mean Error (Meter)	Standard Deviation (Meter)
0°	90. 2%	97. 2%	3.91	2.01
30°	88.9%	81.8%	5.88	2.76
60°	65. 7%	73. 8%	11.47	3.04

4. Conclusions and Future Work

This paper proposes to use Google Maps Street View and image matching to distinguish which side of road the pedestrian is, and estimate the pedestrian positioning result. Firstly, query image is captured by the camera on the smart phone and geo-tagged by the GNSS receiver. The geo-tag is used to determine panoramic images of Google Maps Street View on which area should be downloaded. Then ASIFT is applied to find matches between the query image and the reference images. By decomposing the Homography matrix calculated from the matched features, we can get the rotation-translation matrix and estimate the pedestrian position. Finally, the developed system is tested in an urban environment, and the results indicate that our proposed method improves the correct side rate to 91% and achieves 4-meter positioning performance.

In the future, more experiments will be conducted in different scenarios to verify the performance of the proposed system. In order to make the proposed system more realistic and convenient, we will improve the reliability of our system for different view directions. In addition, the developed positioning algorithm will be extended to a navigation system, which is implemented into wearable smart glasses. The real time processing issue and orientation estimation issue are solved in that stage. Moreover, the accuracy of the panoramic image data provided by Google will also be discussed in the future.

Acknowledgment

This research has gained the permission of Compliance Committee of the University of Tokyo. The authors acknowledge the Grant-in-Aid for Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Oversea Researchers.

References

[1] L.-T. Hsu, Y. Gu, Y. Huang, and S. Kamijo, Urban pedestrian navigation using smartphone-based dead reckoning and 3D maps aided GNSS, IEEE Sensors Journal, 16(5), pp.1281-1293, 2016.

[2] Y. Huang, L.-T. Hsu, Y. Gu, H. Wang, and S. Kamijo, Database calibration for outdoor Wi-Fi positioning system, IEICE Transections on Fundamentals of Electronics, Communications and Computer Sciences, 99(9), pp.1683-1690, 2016.

[3] A. L. Majdik, Y. A. Schoenberg, and D. Scaramuzza. MAV urban localization from Google Street view data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3979-3986, 2013.

[4] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In Proceedings Computer Vision Workshops (ICCV Workshops), pp. 102-109, 2011.

[5] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2599-2606, 2009.

[6] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 255-268, 2010.

[7] P. Agarwal, W. Burgard and L. Spinello. Metric localization using google street view, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3111-3118, 2015.

[8] C.Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, D. and V. Lepetit, Instant outdoor localization and slam initialization from 2.5 d maps, IEEE transactions on visualization and computer graphics, 21(11), pp.1309-1318, 2015.

[9] H. Chu, A. Gallagher, and T. Chen, GPS refinement and camera orientation estimation from a single image and a 2D map. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), pp. 171-178. 2014.

[10] G. Yu and J. M. Morel. ASIFT: An algorithm for fully affine invariant comparison. Image Processing on Line, 1, pp.11-38, 2011.

[11] Google Street View Image API documentation, https://developers.google.com/maps/documentation/streetview/

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions. Image and vision computing, 22(10), pp.761-767, 2004.