IPSJ Transactions on Computer Vision and Applications Vol.6 111–119 (Oct. 2014)

[DOI: 10.2197/ipsjtcva.6.111]

Systems Paper

Mobile Camera Localization Using Aerial-view Images

Hisatoshi Toriya[1,a)] Itaru Kitahara[1,b)] Yuichi Ohta[1]

Received: February 13, 2014, Accepted: July 31, 2014, Released: October 21, 2014

Abstract: This paper proposes a method to estimate a mobile camera's position and orientation by referring to the corresponding points between aerial-view images from a GIS database and mobile camera images. The mobile camera images are taken from the user's viewpoint, and the aerial-view images include the same region. To increase the correspondence accuracy, we generate a virtual top-view image that virtually captures the target region overhead of the user by using the intrinsic parameters of the mobile camera and the inertia (gravity) information. We find corresponding points between the virtual top-view and aerial-view images and estimate a homography matrix that transforms the virtual top-view image into aerial-view image. Finally, the mobile camera's position and orientation are estimated by analyzing the matrix. In some cases, however, it is difficult to obtain a sufficient number of correct corresponding points to estimate the correct homography matrix by capturing only a single virtual top-view image. We solve this problem by stitching virtual top-view images to represent a larger ground region. We experimentally implemented our method on a tablet PC and evaluated its effectiveness.

Keywords: mobile camera localization, keypoint matching, image features, GIS, aerial-view, image stitching

1. Introduction

Over the last few years, mounting Global Positioning Systems (GPSs) on mobile devices, including digital cameras, smartphones, and tablet PCs, has become popular. Many navigation applications have been developed using measured locations. However, using ordinary GPSs is associated with measurement errors of several meters. Since measurement errors are larger in areas with many high-rise buildings, inaccurate information might be presented. For car navigation systems, it is possible to correct the positional measurements by referring to the car's estimated orientation/motion. However, when mobile cameras are handheld, user's walking speeds are much slower than the speed of cars, complicating the application of similar methods used in car navigations. Using an electronic compass is one solution; however, the measurement errors might be several degrees, and the magnetic fields generated by other electronic devices also negatively affect the accuracy of such measurements [1].

Over the last decade, as shown by Google Maps [2], obtaining aerial-view images captured by aircraft has become easy. Each pixel on the aerial-view images of a Geographic Information System (GIS) has global position (latitude and longitude) information.

We propose a localization method that automatically and efficiently estimates the position and orientation of a mobile camera in an outdoor environment by using aerial-view images (**Fig. 1**). First, a mobile camera image is transformed into a virtual top-view image using the information delivered by inertia sensors. The corresponding points between the virtual top-view and aerial-view images are identified by the image features. Finally, the mobile camera's position and orientation are defined by referring to such correspondences.

[1] Graduate School of Systems and Information Engineering, Department of Intelligent Interaction Technologies, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[a)] toriya@image.iit.tsukuba.ac.jp
[b)] kitahara@image.iit.tsukuba.ac.jp

2. Related Works

Mobile camera localization methods are roughly categorized into sensor and image-based methods. Common examples of the former are GPSs, electronic compasses, and inertia sensors. However, all suffer from measurement errors. As an example of the latter category, visual markers are widely used such as ARToolkit [3], which can accurately localize a mobile camera as long as a marker is well observed. Thus, unless we install many markers in the target space, the scenery might be negatively affected. Many methods have been proposed that employ image features rather than visual markers [4], [5]. Even though they demand less from the environment, it is still necessary to prepare landmarks to transform the estimated positions and orientations into a world coordinate system.

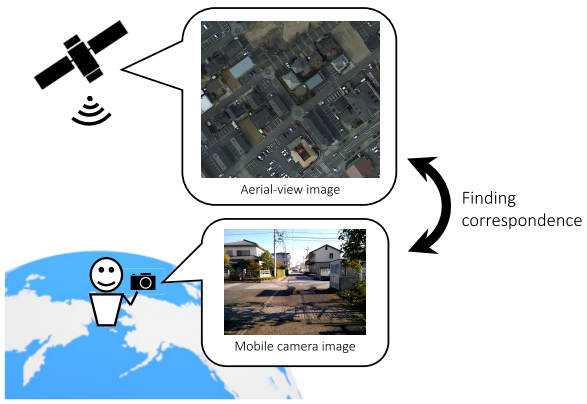


Fig. 1 Localizing user position using correspondence between aerial-view and mobile camera images.

Recently, mobile camera localization methods that use calibrated environmental cameras have been proposed to handle these problems. Tsuru et al. [6] used a mobile camera in combination with calibrated environmental stereo cameras and estimated its position and orientation by capturing the common space. Noda et al. [7] used image sequences from a car-mounted camera and aerial-view images captured by aircraft and extracted road regions from a car camera image and transformed them into aerial views. By searching for correspondences between the transformed road region and the aerial-view image, they computed a homography matrix to estimate the car's position. As long as we surveyed, there have not ever been any localization methods that automatically and efficiently estimate the position and orientation of a mobile camera in an outdoor environment by using aerial-view images.

3. Mobile Camera Localization Using Aerial-view Images

This paper proposes a method to estimate the position and orientation of a mobile camera by extending the above approaches [6], [7]. As a calibrated environmental camera, we focus on aerial-view images of GIS. All the pixels of aerial-view images contain global positional information: latitude and longitude. In a sense, aerial cameras are calibrated in a world coordinate system. Since aerial-view images are captured from a higher viewpoint, the perspective's distortion is small, and we can obtain the visual information of a large area in just a few shots.

Figures 2 and **3** show the processing flow of our proposed method. With a mobile camera, a user takes a landscape picture that partially observes the ground region. Then, we have an assumption that the captured ground area is flat and horizontal (i.e., the surface should be perpendicular to the gravity direction). To reduce the difference in the appearances between the mobile camera and aerial-view images, we generate a virtual top-view image whose viewpoint is virtually transformed to a high viewpoint using the gravity information from an embedded inertia sensor. Using such image features as SIFT (Scale Invariant Feature Transform) [8] keypoint matching allows to identify the corresponding points between the virtual top-views and the aerial-views and to calculate the homography matrix. Finally, we analyze the matrix to localize the mobile camera.

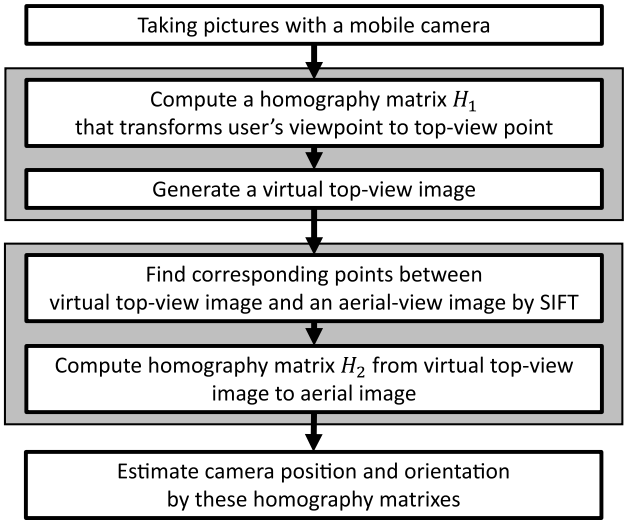


Fig. 2 Processing flow of our proposed method.

4. Generating Virtual Top-view Images

In this section, we describe the process of transforming a mobile camera image to a virtual top-view image (**Fig. 4**). The projection matrices of a mobile camera and a virtual top-view camera is described as follows:

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \tag{1}$$

$$\boldsymbol{P}' = \boldsymbol{K}[\boldsymbol{R}'|\boldsymbol{t}'] \tag{2}$$

with intrinsic camera parameter matrix $\boldsymbol{K}$, rotation matrices $\boldsymbol{R}$ and $\boldsymbol{R}'$, and translation vectors $\boldsymbol{t}$ and $\boldsymbol{t}'$. $\boldsymbol{R}$ and $\boldsymbol{R}'$ are estimated using the gravity information. When we define $\Delta\boldsymbol{R}$ as a coordinate transformation matrix from $\boldsymbol{R}$ to $\boldsymbol{R}'$, homography matrix $\boldsymbol{H}_{\Delta\boldsymbol{R}}$, which projects any points by $\Delta\boldsymbol{R}$, is described as follows:

$$\boldsymbol{H}_{\Delta\boldsymbol{R}} = \boldsymbol{K}(\Delta\boldsymbol{R})\boldsymbol{K}^{-1} \tag{3}$$

$\boldsymbol{H}_{\Delta\boldsymbol{R}}$ is computed using accelerometer data. Translation matrix $\Delta\boldsymbol{t}$ from $\boldsymbol{t}$ to $\boldsymbol{t}'$ is

$$\Delta\boldsymbol{t} = \boldsymbol{t}' - \boldsymbol{t} \tag{4}$$

where $\Delta\boldsymbol{t}$ affects the spatial resolution of the generated virtual top-view image (i.e., the higher that viewpoint $\boldsymbol{t}$ is, the lower the spatial resolution). To make the subsequent image-matching process more efficient, we define $\boldsymbol{H}_{\Delta t}$ to set the spatial resolution of the virtual top-view image almost equal to that of the aerial-view image. Homography matrix $\boldsymbol{H}_1$, which projects the image of the user viewpoint to a virtual top-viewpoint image, is described as:

$$\boldsymbol{H}_1 = \boldsymbol{H}_{\Delta t}\boldsymbol{H}_{\Delta\boldsymbol{R}} \tag{5}$$

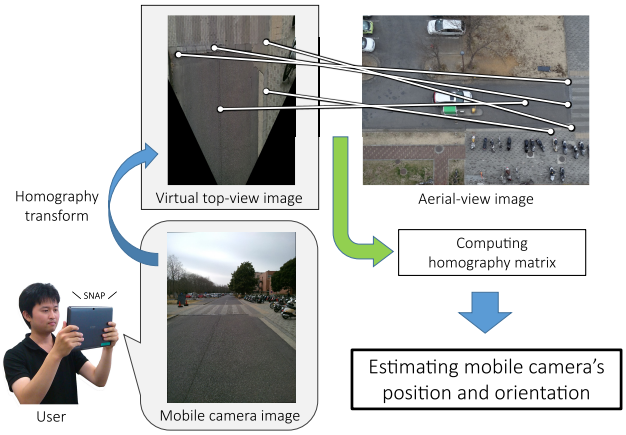


Fig. 3 Mobile camera localization using aerial-view images.

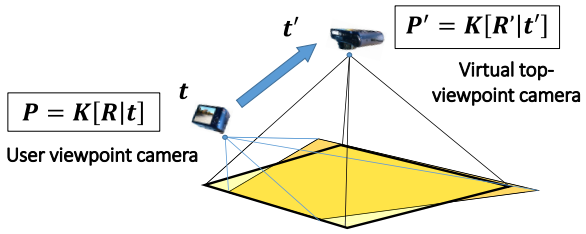


Fig. 4 Generating virtual top-view image.

Fig. 5 Mobile camera image (left) and generated virtual top-view image (right).

Figure 5 shows the result of generating a virtual top-view image.

5. Finding Keypoint Correspondence

In this section, we explain how to determine the correspondence between virtual top-view and aerial-view images to compute homography matrix $\boldsymbol{H}_2$, which projects the virtual top-view image to the aerial-view image. We increase the accuracy of the estimated homography matrix by reducing the false matches and scaling the virtual top-view image.

5.1 Keypoint Matching by SIFT

For the image matching, we use SIFT [8], a keypoint detection and description algorithm. By transforming a mobile camera image to a virtual top-view image, the factors that cause differences in the appearance between the two images are reduced to just scale and illumination. The keypoint features described by SIFT have the advantage of being robust for changes of rotation, illumination, and scale. Therefore, we believe that SIFT is a good choice for image matching (**Fig. 6**).

5.2 Removing False Matches

We identified the corresponding points using SIFT including false matches. As they increase, it becomes difficult to accurately estimate a homography matrix, even if a robust estimation method is employed, such as RANSAC [9]. Therefore, we must reduce the number of false matches on the keypoint orientation and scale.

5.2.1 Keypoint Orientation

The SIFT keypoints include information on the orientation of the gradients. We reduce the false matches using the orientation. First, we generate a histogram of the orientation differences between pairs of corresponding points. We assume that correctly matched pairs have similar orientation differences (i.e., they converge to a few bins of the histogram). As **Fig. 7** shows, when a matching pair lies a certain distance from the highest frequency bin, it is filtered out.

5.2.2 Keypoint Scale

False matches are also reduced using a keypoint scale. SIFT constructs image pyramids that are robust to scale changes. Keypoints have information about the number of image pyramids in

Fig. 6 SIFT matching result.

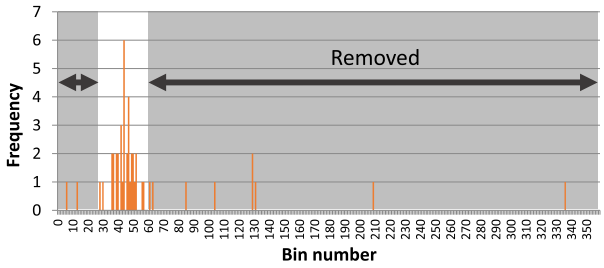


Fig. 7 Removing false matches using histogram of brightness gradient differences.

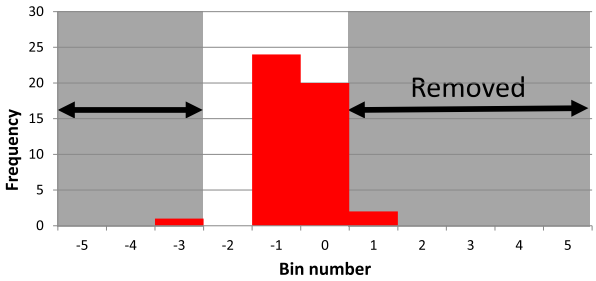


Fig. 8 Removing false matches using histogram of layer number differences.

which a particular keypoint is detected. Here, we generate a histogram that shows the differences in the pyramid numbers between pairs of the corresponding points as well as the histogram of the keypoint orientation described in the previous section. We assume that correctly matching pairs have similar differences. When a matching pair lies a certain distance from the highest frequency bin, it is filtered out (**Fig. 8**).

After the SIFT matching and filtering out of the false matches, the homography matrix is estimated by a robust estimation method called RANSAC by referring to the corresponding points. When use RANSAC, four matches are used at each iteration. **Figure 9** shows a result obtained after removing the false matches.

5.2.3 Fitting Scales

Although SIFT is robust to scale changes, if they are too large, it is difficult to correctly define the correspondence [10]. Therefore, we control the spatial resolution of the virtual top-view image, when its resolution is much higher than that of the aerial-view image. The spatial resolution of the aerial-view image is obtained from the GIS database, and that of the virtual top-view image is computed from the information on the mobile camera's tilt angle, height, and intrinsic parameters. About height parameter, it is difficult to get the parameter automatically, so we should have users input the height parameter. If the spatial resolutions

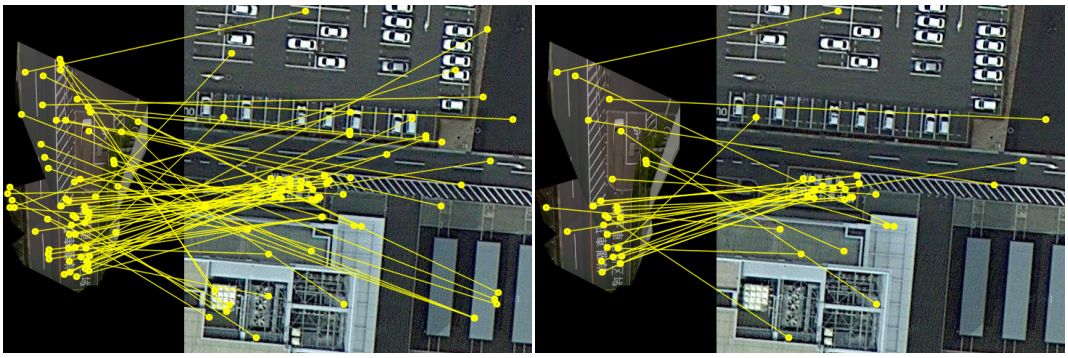


Fig. 9 Before removing (left) and after removing (right).

of a virtual top-view and an aerial-view images are defined as $res_{virtual\_topview}$ and $res_{aerialview}$, a scaling matrix can be described as

$$\boldsymbol{S} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where scaling parameter s is defined as

$$s = \frac{res_{virtual\_topview}}{res_{aerialview}}. \tag{7}$$

6. Mobile Camera Localization

In Sections 4 and 5, we explained how to compute homography matrix $\boldsymbol{H}_1$, which transforms a mobile camera image to a virtual top-view image, and matrix $\boldsymbol{H}_2$, which transforms the virtual top-view image to an aerial-view image. Therefore the following product of these matrices

$$\boldsymbol{H} = \boldsymbol{H}_2\boldsymbol{H}_1 \tag{8}$$

is a homography matrix that transforms mobile camera images to aerial-view images. If a homography matrix from one plane to another is available and mobile camera's intrinsic parameter matrix $\boldsymbol{K}$ is known, we can estimate the mobile camera's 3D position and orientation [3], [11] in the former plane coordinate system.

$\boldsymbol{H}$ in Eq. (8) is a homography matrix from a mobile camera image to an aerial-view image. The homography matrix from an aerial-view image to a mobile camera image is $\boldsymbol{H}^{-1}$. $\boldsymbol{H}^{-1}$ is defined as:

$$\boldsymbol{H}^{-1} \stackrel{\text{def}}{=} [\boldsymbol{h}_1\ \boldsymbol{h}_2\ \boldsymbol{h}_3]. \tag{9}$$

And the rotation matrices and the translation matrix are

$$\boldsymbol{r}_1 = \frac{\boldsymbol{K}^{-1}\boldsymbol{h}_1}{\left\|\boldsymbol{K}^{-1}\boldsymbol{h}_1\right\|} \tag{10}$$

$$\boldsymbol{r}_2 = \frac{\boldsymbol{K}^{-1}\boldsymbol{h}_2}{\left\|\boldsymbol{K}^{-1}\boldsymbol{h}_2\right\|} \tag{11}$$

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2 \tag{12}$$

$$\boldsymbol{t} = \frac{\boldsymbol{K}^{-1}\boldsymbol{h}_3}{\left\|\boldsymbol{K}^{-1}\boldsymbol{h}_1\right\|} = \frac{\boldsymbol{K}^{-1}\boldsymbol{h}_3}{\left\|\boldsymbol{K}^{-1}\boldsymbol{h}_2\right\|}. \tag{13}$$

Then mobile camera's rotation matrix $\boldsymbol{R}$ and position $\boldsymbol{c}$ in the coordinate system of an aerial-view image are described as:

$$\boldsymbol{R} = [\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3] \tag{14}$$

$$\boldsymbol{c} = -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{t}. \tag{15}$$

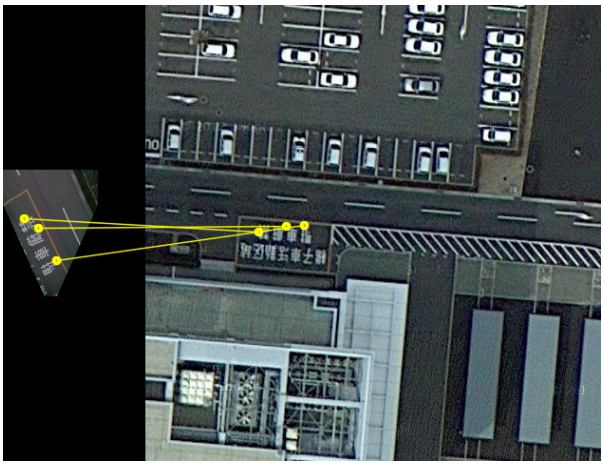


Fig. 10 SIFT matching results between virtual top-view and aerial-view images.

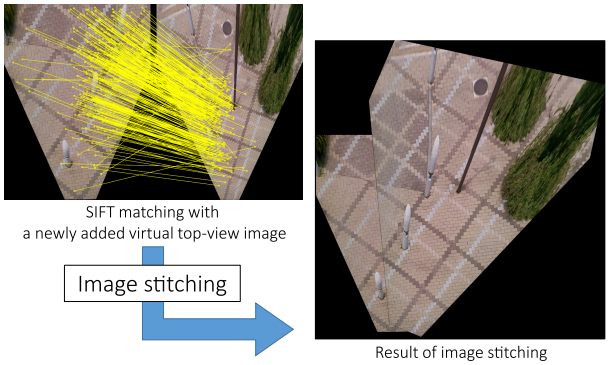


Fig. 11 Stitching virtual top-view images.

7. Generating a Stitched Virtual Top-view Image

Users take pictures while standing on the ground. Sometimes too few ground features are observed in mobile camera images. This complicates obtaining a sufficient amount of keypoint correspondence to localize the mobile camera (**Fig. 10**). To resolve this problem, we generated a virtual top-view image that observes a much larger ground region by stitching multiple virtual top-view images together.

Next we identify the corresponding points between two virtual top-view images, which include a common ground area (**Fig. 11**). Then a homography matrix is computed by referring to these corresponding points, and the latter virtual top-view image of the

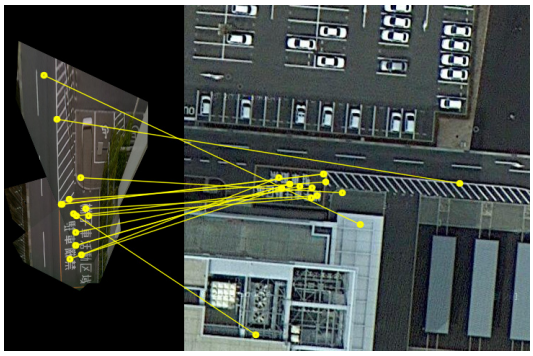


Fig. 12 SIFT matching results between stitched virtual top-view and aerial-view images.

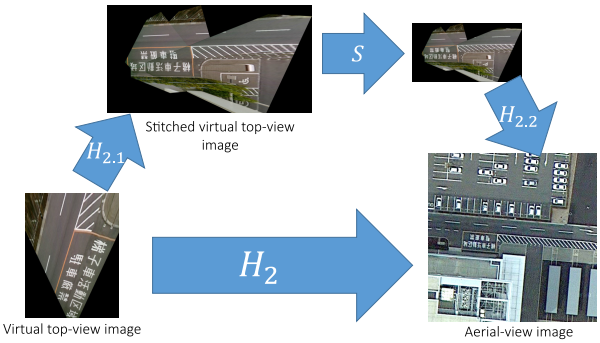


Fig. 13 Computing $\boldsymbol{H}_2$ for image stitching.

multiple images is superimposed on the first image. By repeating this process, a stitched virtual top-view image containing that contains much more visual information on the ground region is generated (**Fig. 12**).

As shown in **Fig. 13**, suppose that $\boldsymbol{H}_{2.1}$ is a homography matrix from a virtual top-view image to a stitched virtual top-view image and $\boldsymbol{H}_{2.2}$ is a homography matrix from a scaled-and-stitched virtual top-view image to an aerial-view image. Then a homography matrix from a virtual top-view image to an aerial-view image is described as follows with $\boldsymbol{S}$ defined in Eq. (6):

$$\boldsymbol{H}_2 = \boldsymbol{H}_{2.2}\,\boldsymbol{S}\,\boldsymbol{H}_{2.1}. \tag{16}$$

8. Experiments

In this section, we introduce two experiments that we conducted to confirm the effectiveness of our proposed mobile camera localization method.

8.1 Experiment 1: Relevance to Spatial Resolution of Aerial-view Images

In this experiment, our aim is to acquire knowledge about spatial resolution to realize accurate mobile camera localization. To quantitatively control the spatial resolution, we captured aerial-view images using an unmanned aerial vehicle (UAV) and generated an aerial-view image dataset by decreasing the resolution.

8.1.1 Experimental Environment, Equipment, and Dataset

We captured original aerial-view images with a UAV (Cinestar 6 by FREEFLY SYSTEMS [12]) at an open space at the University of Tsukuba (**Fig. 14**). A single-lens reflex (SLR) camera (Canon EOS 6D with an EF4028STM Canon lens) was embedded on the UAV, and aerial-view images were captured at 20 meters

Fig. 14 UAV used for experiment 1.

Fig. 15 Aerial-view image captured in experiment 1.

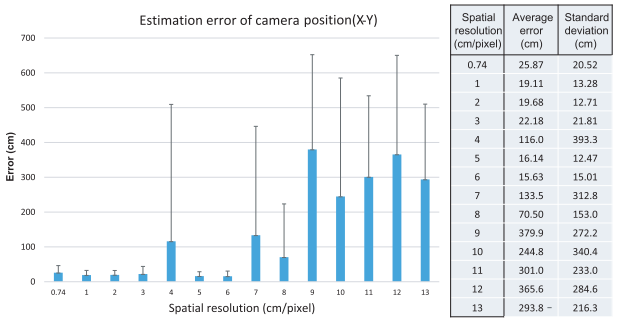
Fig. 16 Capturing mobile camera images.

height. An example of a captured aerial-view image is shown in **Fig. 15**. The image size is 5,472 × 3,648 pixels, and the spatial resolution is about 0.74 cm/pixel. We generated an aerial-view image dataset with 14 levels of spatial resolution: 0.74, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0 cm/pixel. For reference, the aerial-view images generally obtained from Google Map [2] have about 12.5 cm/pixel spatial resolution around the center of a large city.

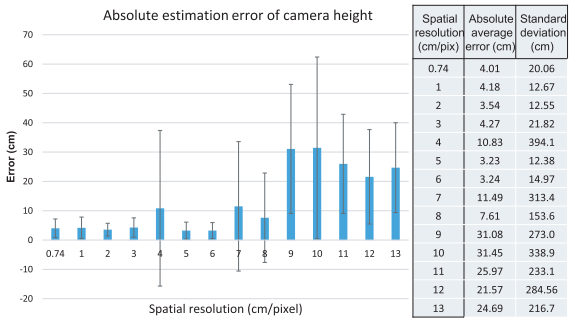
For capturing mobile images, we used an ASUS Pad TF700T tablet PC (OS: Android ver4.2.1) that included both GPS and an accelerometer. The tablet PC (mobile camera) was mounted on a tripod at 150 cm (**Fig. 16**). The angle of depression was 5 degrees. 34 pictures were captured at four positions. **Figure 17** shows the dataset, and **Fig. 18** shows the capturing position and orientation. Based on the acceleration value, we generated virtual top-view images. The spatial resolution of the virtual top-view images was set as the same spatial resolution as the aerial-view images.

8.1.2 Experimental Results

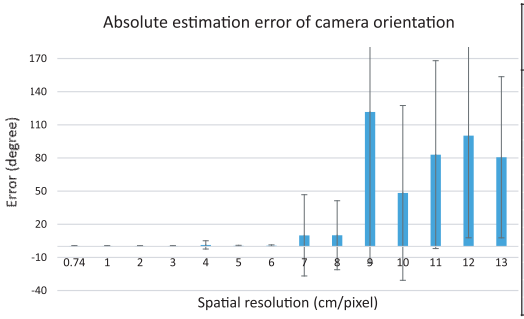
Figure 19 shows the number of mobile images in which the ho-



Spatial resolution (cm/pixel)	Average error (cm)	Standard deviation (cm)
0.74	25.87	20.52
1	19.11	13.28
2	19.68	12.71
3	22.18	21.81
4	116.0	393.3
5	16.14	12.47
6	15.63	15.01
7	133.5	312.8
8	70.50	153.0
9	379.9	272.2
10	244.8	340.4
11	301.0	233.0
12	365.6	284.6
13	293.8	216.3



Spatial resolution (cm/pix)	Absolute average error (cm)	Standard deviation (cm)
0.74	4.01	20.06
1	4.18	12.67
2	3.54	12.55
3	4.27	21.82
4	10.83	394.1
5	3.23	12.38
6	3.24	14.97
7	11.49	313.4
8	7.61	153.6
9	31.08	273.0
10	31.45	338.9
11	25.97	233.1
12	21.57	284.56
13	24.69	216.7



Spatial resolution [cm/pixel]	Absolute average error [degree]	Standard deviation [degree]
0.74	0.40	0.5062
1	0.38	0.5035
2	0.36	0.4877
3	0.39	0.5002
4	1.33	3.8039
5	0.51	0.6306
6	0.64	0.9549
7	9.99	33.95
8	10.09	32.53
9	121.69	56.76
10	48.39	58.18
11	83.02	91.13
12	100.26	102.8
13	80.66	100.9

Fig. 20 Estimation errors of camera position (top left); camera height (top right); camera orientation (bottom).

Fig. 17 Captured mobile camera images for experiment 1.

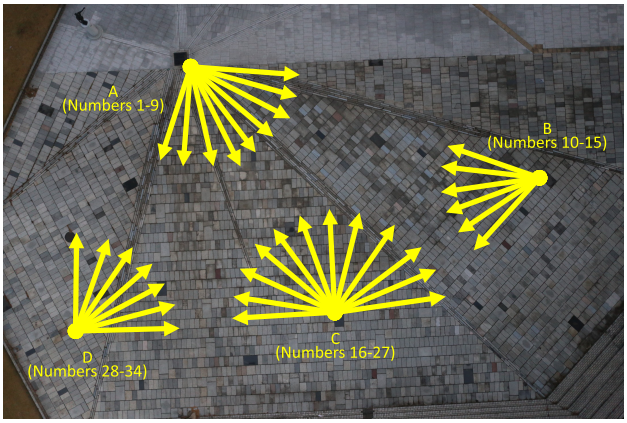


Fig. 18 Captured position and orientation.

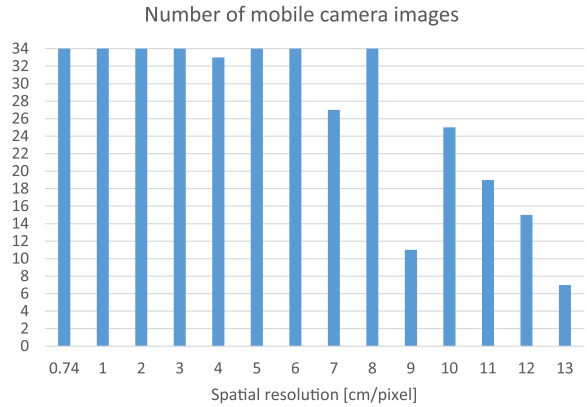


Fig. 19 Number of mobile images in which homography matrix was successfully estimated.

mography matrix was successfully estimated; if we can estimate the homography matrices for all captured mobile camera images, the number is 34. The horizontal axis represents the spatial resolution of the aerial-view images. Computing the homography matrix tends to fail when the spatial resolution of aerial-view images is lower than 8.0 cm/pixel.

Figure 20 shows the estimation error, which was calculated as the average value given by the mobile camera images in which the homography matrix was successfully calculated. Our experimental results confirm that we can realize less than 1.0-meter positional estimation error and less than 1.0-degree orientation error when the spatial resolution exceeds 6.0 cm/pixel. Also, considering Fig. 19, we conclude that our proposed method enables accurate and stable mobile camera localization when the aerial-view image's spatial resolution exceeds 6.0 cm/pixel.

The required spatial resolution of the aerial-view image depends on two items. One is the complexity of the appearance of the captured ground region, and the other is the spatial res-

Fig. 21 Differences in appearance caused by spatial resolution: 12 cm/pixel (left), 6 cm/pixel (center), 1 cm/pixel (right).

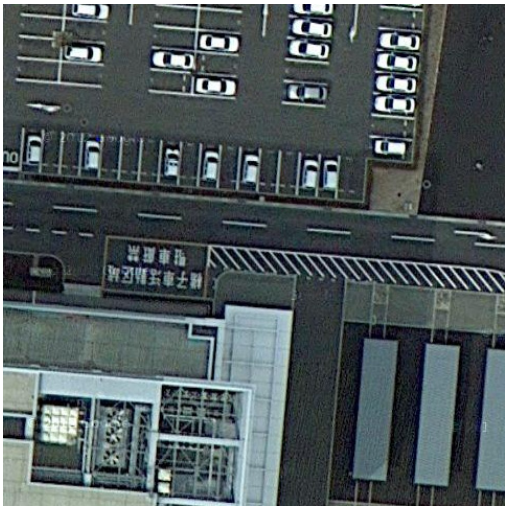


Fig. 22 Aerial-view image for experiment 2.

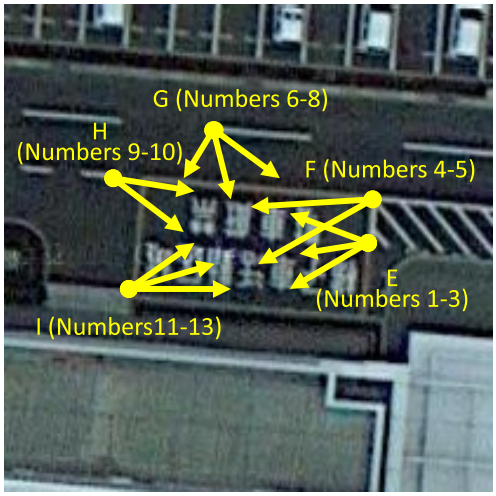


Fig. 23 Captured positions and their orientations.

olution of the mobile camera image. Since the mobile camera image's spatial resolution is usually much higher than the aerial-view image, we only need to consider the relationship between the required spatial resolution of the aerial-view image and the complexity of the appearance of the captured ground region.

Figure 21 shows the appearance of the ground region with three spatial resolutions; 12 cm/pixel, 6 cm/pixel, and 1 cm/pixel. In this experimental environment, the ground region was covered with 30×60 cm stone tiles. If the spatial resolution of the aerial-view image is less than 6 cm/pixel, the smaller side of the tiles is observed with a smaller than the minimum SIFT keypoint. As a result, it is difficult to accurately estimate the correspondence between the images. The accuracy of the mobile camera localization was also degraded.

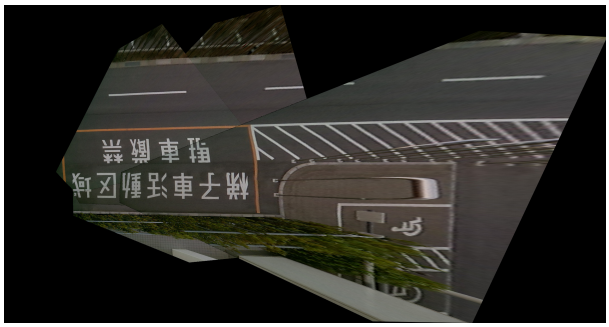


Fig. 24 Stitched virtual top-view image.

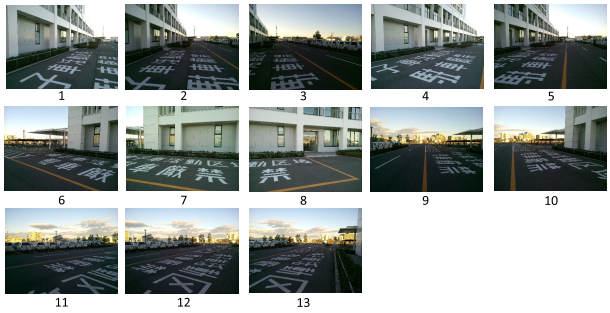


Fig. 25 Mobile camera images for experiment 2.

8.2 Experiment 2: Evaluating Mobile Camera Localization with General Aerial-view Images

As the experimental result in the previous section, the spatial resolution of the aerial-view image given by a general GIS dataset such as Google Maps is not sufficient for our method. To realize a practical method using a general dataset, we generated a stitched virtual top-view image by the stitching process described in Section 7. In this section, we evaluate the usefulness of our method.

8.2.1 Experimental Environment, Equipment, and Dataset

We conducted an experiment around Tsukuba City in Ibaraki prefecture. **Figure 22** shows an aerial-view image whose spatial resolution is 12.5 cm/pixel, and **Fig. 23** shows the captured location and orientation.

We used the same tablet PC as in Experiment 1 and captured several pictures changing camera's position and orientation freely to obtain a stitched virtual top-view image (**Fig. 24**) because the aerial-view images from Google Maps have low spatial resolution. The tablet PC was mounted on a tripod at 150 cm (Fig. 16). The angle of depression was 5 degrees. We took 13 pictures (**Fig. 25**).

8.2.2 Experimental Results

The experiment's estimation error is shown in **Fig. 26**. Even though the spatial resolution of the aerial-view images is low, our proposed method realized averages of 158.8 cm positional estimation error and 1.286 degrees of orientation estimation error by applying the stitching process.

9. Implement a Pilot System

In this section, we describe how realizes the proposed method using available datasets, devices and so on, and discuss the remained difficulty and the solution.

Figure 27 shows a block diagram of our proposed method implemented on a tablet (ASUS Pad TF700T (Android ver4.2.1

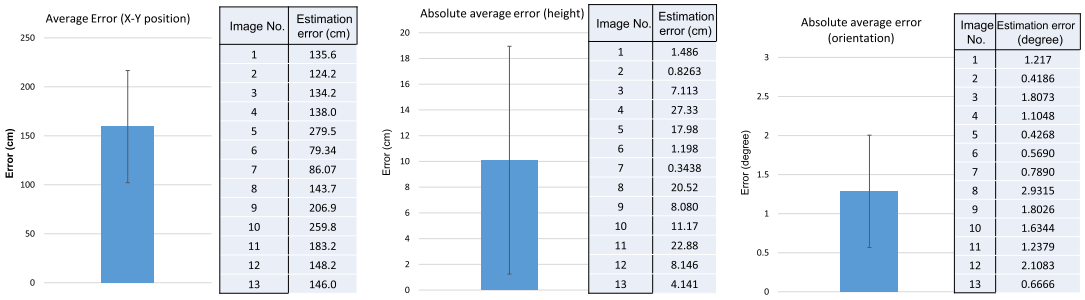


Image No.	Estimation error (cm)
1	135.6
2	124.2
3	134.2
4	138.0
5	279.5
6	79.34
7	86.07
8	143.7
9	206.9
10	259.8
11	183.2
12	148.2
13	146.0

Image No.	Estimation error (cm)
1	1.486
2	0.8263
3	7.113
4	27.33
5	17.98
6	1.198
7	0.3438
8	20.52
9	8.080
10	11.17
11	22.88
12	8.146
13	4.141

Image No.	Estimation error (degree)
1	1.217
2	0.4186
3	1.8073
4	1.1048
5	0.4268
6	0.5690
7	0.7890
8	2.9315
9	1.8026
10	1.6344
11	1.2379
12	2.1083
13	0.6666

Fig. 26 Estimation error of camera position; average error is 158.8 cm, with standard deviation 57.22 cm (left). Estimation error of camera height; average error is 10.09 cm with standard deviation 8.85 cm (middle). Estimation error of camera orientation; average error is 1.286 degrees, with standard deviation 0.719 degrees (right).

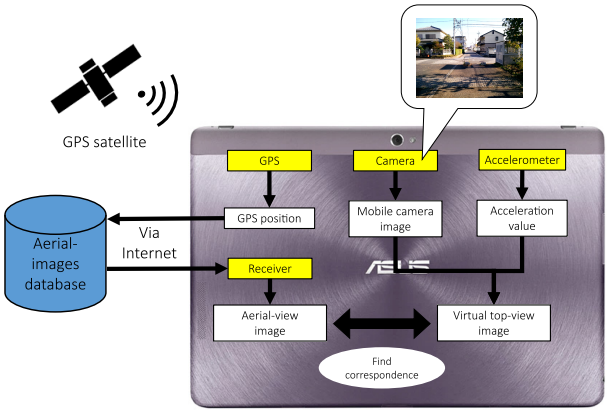


Fig. 27 Block diagram of our localization system.

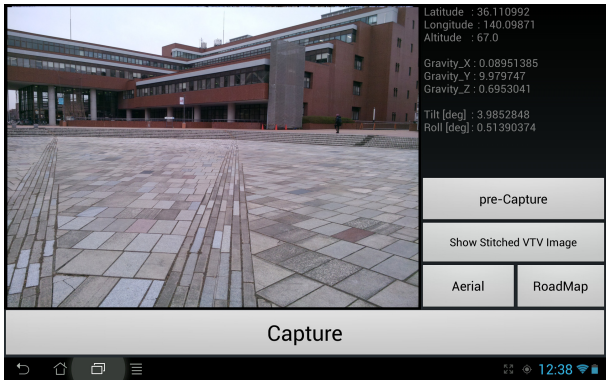


Fig. 28 Display of application system.

Fig. 29 Using our implemented system.

OS)) that included both GPS and an accelerometer. In order to scale the system to localization in a wider range of areas, we clip out the area of interest (around a user) from the global aerial-view images according to the GPS measurement value. This mobile camera accesses the Internet by WiFi to download aerial-view images around it from the GIS dataset. The development tool for image processing is OpenCV4Android SDK ver.2.4.3 [13], and for downloading aerial-view images we used Google Static Maps API ver.2 [14], which sends HTTP requests including GPS measurement values to get aerial-view images from the Google Maps database. The camera height parameter is fixed at 150 cm.

Our application's display is shown in **Fig. 28**. When the "Capture" button is touched, a mobile camera image is captured and a virtual top-view image is generated. Aerial-view images are downloaded from the GIS dataset based on the GPS measurement. Then the corresponding points between the virtual top-view and aerial-view images are estimated. If the stitching process is required, in other words, if the number of corresponding points is insufficient, the user touches the "pre-Capture" button several times to change the viewpoint and a stitched virtual top-view image is generated. He can check the generated stitched virtual top-view image by touching the "Show Stitched VTV Image" button.

When keypoint matching is successful, a homography matrix from the mobile camera image to the aerial-view image is computed. In this application, with the homography matrix, we can overlap other images on the captured image. By touching the "Aerial" button, an aerial-view image projected onto the mobile camera viewpoint is overlapped, and when the "RoadMap" button is touched, the same process is applied to a road map image. **Figure 29** shows a scene of our implemented system in an outdoor environment.

One of the difficulties is reducing computational time. SIFT needs much computational cost, which is too heavy for ordinal tablet PCs. One solution is to send the captured image and sensor data to a PC that has higher-computational power, and reply the commutating result to the tablet PC. Another solution is to replace SIFT with other faster keypoint descriptors such as ORB [15] or FREAK [16].

10. Conclusion and Future Works

In this paper, we proposed a method for estimating the posi-

tion and orientation of a mobile camera by referring to the corresponding points between mobile camera and aerial-view images. We implemented our proposed method on an Android tablet PC. In two experiments, we confirmed that our proposed method enables more accurate localization of mobile camera than such sensor-based localization as GPSs or electronic compasses. Future works will be to improve the accuracy of keypoint detection and matching by conducting more experiments in various environments, including non-planar shaped ground surfaces.

References

[1] Yun, X., Bachmann, E.R., McGhee, R.B., Whalen, R.H., Roberts, R.L., Knapp, R.G., Healey, A.J. and Zyda, M.J.: Testing and evaluation of an integrated GPS/INS system for small AUV navigation, *IEEE Journal of Oceanic Engineering*, Vol.24, No.3, pp.396–404 (1999).

[2] Google Maps, available from ⟨https://maps.google.co.jp/⟩ (accessed 2013-12-02).

[3] Kato, H. and Mark, B.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferecing System, *Proc. 2nd International Workshop on Augmented Reality* (*IWAR*) (1999).

[4] Gordon, I. and Lowe, D.G.: Scene Modeling, Recognition and Tracking with Invariant Image Features, *ISMAR* (2004).

[5] Klein, G. and Murray, D.: Parallel tracking and mapping for small AR workspace, *ISMAR* (2007).

[6] Tsuru, H., Kitahara, I. and Ohta, Y.: A Mobile Camera Calibration Method Using an Environmental Stereo Camera, *IEEJ Trans. Electronics, Information and Systems*, Vol.133, No.1, pp.47–53 (2013).

[7] Noda, M., Takahashi, T., Deguchi, D., Ide, I., Murase, H., Kojima, Y. and Naito, T.: Vehicle ego-localization by matching in-vehicle camera images to an aerial image, *Computer Vision-ACCV 2010 Workshops* (2011).

[8] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision* (*IJCV*), pp.91–110 (2004).

[9] Fisher, M.A. and Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.24, pp.381–395 (1981).

[10] Azad, P., Asfour, T. and Dillmann, R.: Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS 2009*) (2009).

[11] Zhang, Z.: A Flexible New Technique for Camera Calibration, Technical report MSR-TR-98-71, Microsoft Research (Dec. 1998).

[12] FREEFLY SYSTEMS, available from ⟨http://www.freeflysystems.com/⟩ (accessed 2013-12-25).

[13] OpenCV, available from ⟨http://opencv.org/⟩ (accessed 2013-12-02).

[14] Google Static Maps API, available from ⟨https://developers.google.com/maps/documentation/staticmaps/⟩ (accessed 2013-12-02).

[15] Ethan, R., Rabaud, V., Konolige, K. and Bradski, G.: ORB: An efficient alternative to SIFT or SURF, *2011 IEEE International Conference on Computer Vision* (*ICCV*) (2011).

[16] Alahi, A., Raphael, O. and Vandergheynst, P.: Freak: Fast retina keypoint, *2012 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*) (2012).

Hisatoshi Toriya received his B.S. in Systems Engineering from University of Tsukuba, Japan in 2012. He is a master's candidate in the Department of Intelligent Interaction Technologies, the Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests are computer vision.

Itaru Kitahara received his B.E. and M.E. degrees in Science Engineering from University of Tsukuba, Japan in 1994 and 1996, respectively. In 1996, he joined Sharp Corporation. 2000–2003, he was a research associate of the Center for Tsukuba Advanced Research Alliance, University of Tsukuba. He received his Ph.D. in 2003. 2003–2005, he was a researcher at ATR. 2005–2008, he was an assistant professor at University of Tsukuba. Since 2008, he has been an associate professor at University of Tsukuba. His research interests include computer vision, mixed reality, and intelligent image media.

Yuichi Ohta received his B.E. and M.E. degrees in Engineering from Kyoto University, Japan in 1972 and 1974, respectively. He received his Ph.D. from Kyoto University in 1980. 1978–1981, he was a Research Associate of Kyoto University. 1981–1987 Assistant Professor at University of Tsukuba. 1987–1992 Associate Professor and 1992–2004 Professor at University of Tsukuba. 2004–2009 Professor in the Graduate School of University of Tsukuba. 2009–2012 he served as Provost of the Graduate School. Since 2013, he is Vice President of University of Tsukuba. 1982–1983, he was Visiting Scientist in Computer Science Department, Carnegie Mellon University. He is an IAPR Fellow, an IEICE Fellow, and an IPSJ Fellow.

(Communicated by *Cheng-Lin Liu*)