

# A Space-Mapping Method for Object Location Estimation Adaptive to Camera Setup Changes for Vision-Based Automation Applications

Chih-Jen Wu, *Member, IEEE*, and Wen-Hsiang Tsai, *Senior Member, IEEE*

**Abstract**—A new space-mapping method for object location estimation, which is adaptive to camera setup changes, for use in various automation applications is proposed. The location of an object appearing in an image is estimated by mapping image coordinates of object points to corresponding real-world coordinates using a mapping table, which is constructed in two stages, with the first stage for establishing a basic table using bilinear interpolation in the camera manufacturing environment and the second for adapting the table to changes of camera heights and orientations in the application field. Analytic equations for table adaptation are derived by skillful utilization of both image formation and camera geometry properties. Good experimental results are shown to prove the feasibility of the proposed method.

**Index Terms**—Camera setup changes, object location estimation, space mapping, table adaptation.

## I. INTRODUCTION

Video cameras are used in various automation applications, including automatic estimations of object locations in indoor environments using object images acquired by cameras affixed to walls or ceilings. A conventional solution to the object location estimation problem is to conduct camera calibration to obtain a set of camera parameters, followed by using the parameters to compute the object location [1]–[6]. Most camera calibration methods use landmarks to compute camera parameters [4], [5], and the process is generally complicated. A flexible method based on this approach is proposed recently by Zhang [6], which only requires the camera to observe a planar pattern shown in a few different orientations. An alternative approach to object location estimation is to use a space-mapping *table* [7]–[10] which transforms the image

space into the real-world space, thus avoiding the work of computing camera parameters. The table is constructed with the aid of a calibration pattern before the camera is deployed in an application environment.

The space-mapping-based approach to object location estimation, however, is sensitive to camera setup changes. That is, after a space-mapping table is constructed for a specific camera setup according to a certain camera-environment configuration, the camera should be used in identical configurations thereafter; otherwise, the table will not work. This weakness causes inconvenience in using the camera in various application fields.

To solve such a camera-setup sensitive problem for the space-mapping approach, one way is to construct a new table for every new camera-environment configuration. But this is often difficult to carry out after the camera is delivered to a user who does not know the mapping table construction process or/and has no calibration pattern for use in the field. In this letter, we investigate the possibility of *automatically* modifying the original space-mapping table for use in the new environment. Note that this problem of adapting the space-mapping table to new camera setups has not been studied so far. The camera is assumed to be *general* in type with a *fixed* focal length and affixed to a ceiling. In case sharpness of taken images is concerned, a wide-angle camera with a small hyperfocal length of just a few inches may be used, which always takes sharp scene images at distances beyond a half of the hyperfocal length.

We first describe the idea and the details of the proposed method in Sections II and III, respectively. Some experimental results are given in Section IV, followed by conclusions in Section V.

## II. IDEA OF PROPOSED METHOD

The proposed method includes two stages, one conducted in an in-factory environment and the other in an in-field one. The details are described in the following algorithm. See Fig. 1 for an illustration.

Note that in the above algorithm, we assume that the processed object feature point in Step 7 is on the ground.

## III. BASIC SPACE-MAPPING TABLE CONSTRUCTION AND MODIFICATIONS

As mentioned in Step 4 of Algorithm 1, a quadrilateral mapping technique has been designed in this letter to perform

Manuscript received May 15, 2009; revised January 9, 2010; accepted March 28, 2010. Date of publication May 19, 2011; date of current version January 6, 2012. This work was supported by the Ministry of Economic Affairs, under Project MOEA 97-EC-17-A-02-S1-032 in Technology Development Program for Academia. A preliminary version of this paper appeared in “Knowledge-based and intelligent information and engineering systems, in Proceedings of the KES, Sep. 2009. This paper was recommended by Associate Editor T. Zhang.

C.-J. Wu is with Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: gis91813@cis.nctu.edu.tw).

W.-H. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan, and also with the Department of Information Communication, Asia University, Taichung 41354, Taiwan (e-mail: whtsai@asia.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2011.2157196

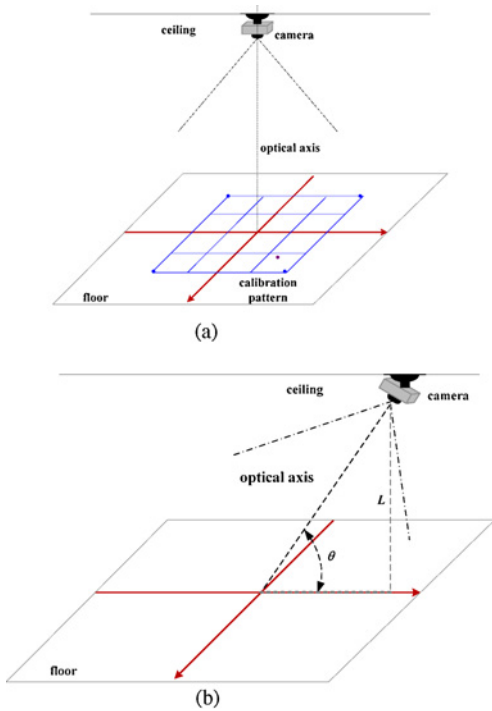


Fig. 1. Illustration of camera setup. (a) Construction of space-mapping table in Stage 1 of proposed method. (b) Camera setup changed to be with a new height of  $L$  and an additional tilt angle of  $\theta$ .

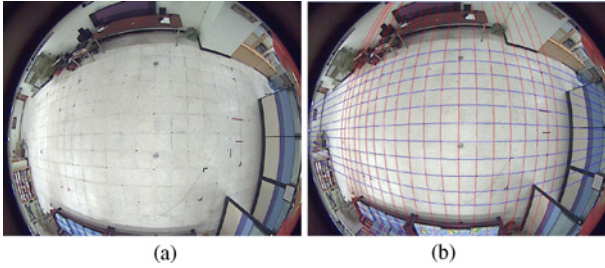


Fig. 2. Illustration of quadrilateral shape extraction from a grid pattern on floor. (a) Image of the grid pattern. (b) Lines approximating the grid lines.

the construction of a space-mapping table  $T$  by two steps: 1) finding pairs of corresponding quadrilaterals, one in the calibration pattern in the image and the other in the real world, and 2) transforming the image and real-world coordinates of corresponding points within the quadrilaterals using a bilinear interpolation technique, as illustrated by Figs. 2 and 3. The details are omitted due to the page limit.

After the basic table is obtained with the camera affixed to a ceiling at a certain height  $H_0$  with respect to a floor  $F_0$ , if the camera is used later at a different ceiling height  $H_1$  with respect to a second floor  $F_1$ , then the table is no more applicable and table content modification is necessary, which we call ceiling-height adaptation, as mentioned in Step 5 in Algorithm 1. To do this, first note that an image point  $p$  is formed in principle by *any* of the real-world points which all lie on a light ray  $R$  going into the camera's lens and then onto the image plane. As illustrated in Fig. 4(a), suppose that this light ray  $R$  intersects floor  $F_0$  at  $P_0$  and floor  $F_1$  at  $P_1$ . If the image coordinates of  $p$  are  $(u, v)$ , then the real-world coordinates  $(x_0, y_0)$  in the basic table corresponding to  $(u,$

**Algorithm 1** Object location estimation by space-mapping table construction and modification.

*Stage 1. Construction of a basic space-mapping table in the factory [see Fig. 1(a)].*

- Step 1) Affix the camera to the ceiling at a certain height  $H_0$  with the optical axis of the camera pointing to the floor perpendicularly.
- Step 2) Place a calibration pattern  $O$  on the ground right under the camera, take an image of it, extract relevant feature points from the image, and find the coordinates of them.
- Step 3) Measure the real-world coordinates of those feature points in the calibration pattern which correspond to the extracted feature points in the image.
- Step 4) *Coordinate mapping*: Use a quadrilateral mapping technique to construct a *basic space-mapping table*  $T$ , which maps each image coordinate pair  $(u_i, v_j)$  to a real-world coordinate pair  $(x_{ij}, y_{ij})$ .

*Stage 2. Modifying the basic table for a new environment [see Fig. 1(b)].*

- Step 5) *Ceiling-height adaptation*: If the new in-field camera setup to be carried out includes *just* a change of the original ceiling height  $H_0$ , perform the following operations to modify the basic table  $T$ ; else, go to the next step.

5.1 Affix the camera to the ceiling and measure the new ceiling height  $H_1$ .

5.2 With  $H_1$  as input, modify table  $T$  into a new one  $T_1$  by a technique of ceiling height adaptation (described later in the next section), and go to Step 7.

- Step 6) *Camera-orientation adaptation*: Perform the following operations to modify the basic table  $T$ .

6.1 Affix the camera to the ceiling, and measure the ceiling height  $L$  and the camera's orientation  $q$ .

6.2 With  $L$  and  $\theta$  as input, modify table  $T$  to be a new one  $T_1$  by a technique of camera orientation adaptation (described later in the next section).

- Step 7) *Location estimation*: Locate an object  $B$  in the real-world space using  $T_1$  in the following way.

7.1 Acquired an image  $I$  of  $B$  with the camera.

7.2 Detect  $B$  in  $I$  and find a feature point  $p$  on it with coordinates  $(u, v)$ .

7.3 Use  $(u, v)$  to look up  $T$  to get the real-world coordinates  $(x, y)$  of the real-world point  $P$  corresponding to  $p$  as the desired object location result.

$v$ ) actually are those of  $P_0$  on  $F_0$  instead of being the desired ones,  $(x_1, y_1)$ , of  $P_1$  on  $F_1$ . To correct this error, we derive first the equalities according to the concept of *side proportionality* in a triangle as follows:

$$x_1 = x_0 \frac{H_1}{H_0}; \quad y_1 = y_0 \frac{H_1}{H_0}. \quad (1)$$

That is, the table lookup result  $(x_0, y_0)$  corresponding to the image coordinates  $(u, v)$  of a real-world point  $P_1$  on  $F_1$  should

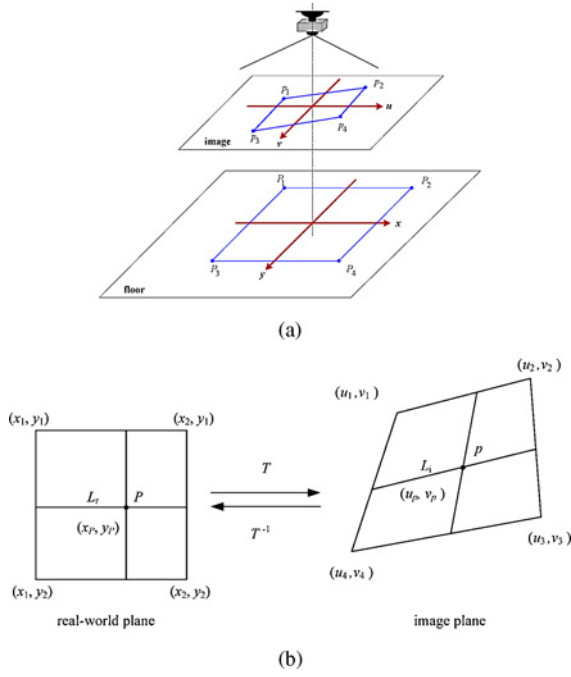


Fig. 3. Quadrilateral mapping. (a) Mapping of corresponding quadrilaterals in the image and the calibration pattern. (b) Location estimation of a space point by inverse bilinear interpolation.

be magnified in proportion to  $H_1/H_0$  to be  $(x_1, y_1)$  as the desired result. Note that we assume here: 1) the real-world coordinate system  $x - y - z$  is set up at the camera's lens center; 2) the optical axis is taken to be the  $z$ -axis; and 3) the location of object point  $P$  described by  $(x_1, y_1)$  is measured with respect to this system.

Now, assume that the camera is affixed to the ceiling differently with a tilt angle of  $\theta$  and a height of  $L$  with respect to floor  $F_1$ , as shown in Fig. 4(b). Here, the location of object point  $P_1$  on  $F_1$  to be estimated is specified by the real-world coordinates  $(x_1, y_1)$  with respect to the downward projection point  $O$  of the camera's lens center onto  $F_1$ , where the  $x$ -axis is assumed to be coincident with the projection of the camera's optical axis on  $F_1$ . Let the coordinates of  $P_1$  in the acquired image be  $(u, v)$ . Again the basic table is inapplicable here; the table lookup results, namely, the real-world coordinates  $(x_0, y_0)$ , are actually those of a real-world point on a floor  $F_0$  at a distance  $H_0$  to the camera's lens center, instead of being the desired real-world coordinates  $(x_1, y_1)$  of  $P_1$  on  $F_1$ . Again, table modification is necessary here, which is called camera orientation adaptation in Step 6 of Algorithm 1.

To correct the values  $(x_0, y_0)$  into  $(x_1, y_1)$ , we rotate  $F_1$  through an angle of  $90^\circ - q$  with  $P_1$  as the rotation pivot point, such that the resulting plane  $F_1'$  becomes perpendicular to the camera's optical axis and the lateral view of the rotation result seen from the positive  $y$ -axis direction becomes the one shown in Fig. 5. The original floor  $F_0$  is also shown in the figure. Assume that the distance of  $P_1$  on  $F_1'$  to the camera's optical axis is  $x'$ . Then, according to the concept of side proportionality again, we have

$$\frac{x_0}{x'} = \frac{H_0}{H_1}. \quad (2)$$

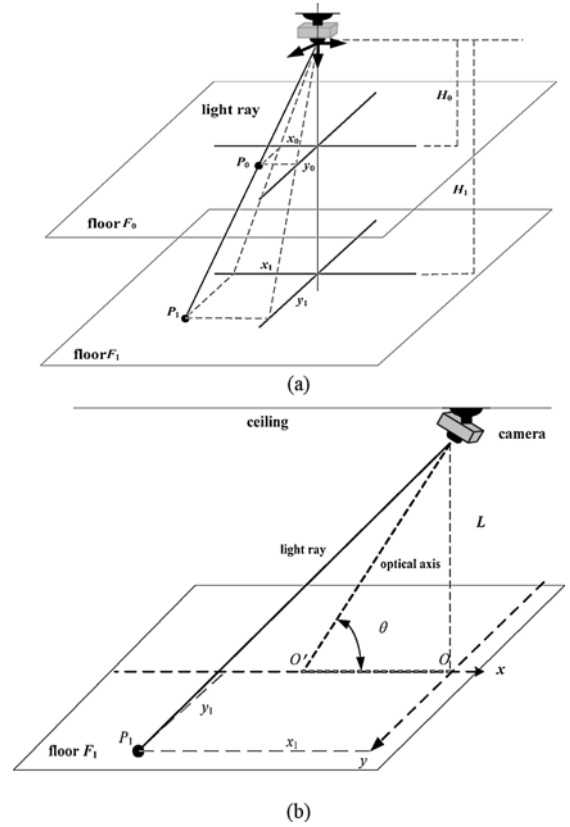


Fig. 4. Illustration of camera setup changes. (a) Use of side proportionality to compute coordinates of point  $P_1$  on a floor  $F_1$  with a ceiling height  $H_1$ . (b) Camera with tilt angle  $\theta$ .

Also, by geometry and trigonometry we have

$$\sin \theta = \frac{x'}{M} \quad (3)$$

$$\sin \theta = \frac{L}{N + H_0} \quad (4)$$

$$\cos \theta = \frac{x_1 - M}{N + H_0} \quad (5)$$

$$\cos \theta = \frac{H_1 - (N + H_0)}{M}. \quad (6)$$

From (4) and (5), we get  $N + H_0 = L/\sin \theta = (x_1 - M)/\cos \theta$ , or equivalently

$$(x_1 - M) \sin \theta = L \cos \theta. \quad (7)$$

Also, from (2) and (3), we get  $x_0/\sin \theta = MH_0/H_1$ , or equivalently

$$H_1 = \frac{H_0 M \sin \theta}{x_0}. \quad (8)$$

From (4), (6), and (8), we get

$$M = \frac{L}{\sin \theta} \times \frac{x_0}{H_0 \sin \theta - x_0 \cos \theta}. \quad (9)$$

And from (7) and (9), we get one of two desired coordinates

$$x_1 = L \times \frac{H_0 \cos \theta + x_0 \sin \theta}{H_0 \sin \theta - x_0 \cos \theta}. \quad (10)$$

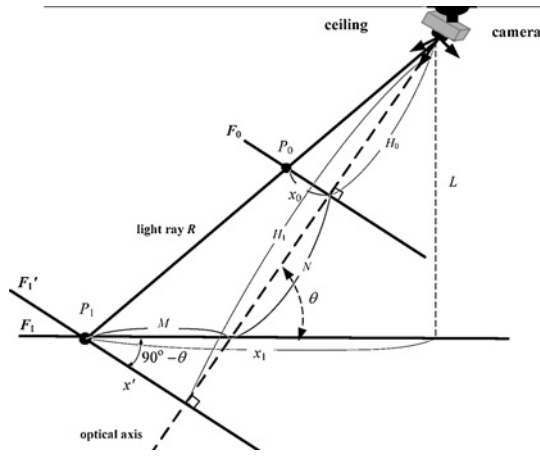


Fig. 5. Lateral view from positive  $y$ -axis direction of the rotation result of floor  $F_1$  in Fig. 4(b) through an angle of  $90^\circ - \theta$  with  $P_1$  as the rotation pivot point.

On the contrary, because the  $x$ -axis on  $F_1$  is assumed to be coincident with the projection of the camera's optical axis on  $F_1$  and because the rotation of  $F_1$  into  $F_1'$  is pivoted in the  $y$ -direction, we have  $y' = y_1$ . Also, according to (1) we have  $y'y_0 = H_1/H_0 = x'x_0$ . Therefore,  $y_1 = y' = y_0(x'/x_0)$ , from which and (3) and (9), we get the other desired coordinate

$$y_1 = L \times \frac{y_0}{H_0 \sin \theta - x_0 \cos \theta}. \quad (11)$$

#### IV. EXPERIMENTAL RESULTS

A series of experiments using a fish-eye camera shown in Fig. 6(a) have been conducted to test the precision of the proposed method for object location estimation. The camera was attached to a rotator connected to a rod with an adjustable length, and so can be tilted arbitrarily and raised to any height. An image taken with the camera looking downward is shown in Fig. 2. We show additionally three images in Fig. 6(b)–(d) among those taken with the camera in two distinct categories of setups: 1) looking downward at the heights of 200, 225, and 250 cm, and 2) being tilted for  $90^\circ$ ,  $70^\circ$ , and  $50^\circ$  at the height of 200 cm. The images are all of the resolution of  $1280 \times 1024$ .

The first case of Category-A setups at the height of 200 cm was regarded as the *original* camera setup for building a basic space mapping table. Specifically, after the image of Fig. 6(b) was taken with the downward-looking camera affixed at the height of 200 cm, all the grid points in the image were extracted to get their image coordinates, forming a set  $I_c$ . Also, the real-world coordinates of each grid point were measured manually to form a set  $W_c$ . The two sets  $I_c$  and  $W_c$  were then used to construct a basic space-mapping table  $T$ .

To test the precision of Table  $T$ , nine nongrid points among the grid ones, which also appear in Fig. 6(b), were selected, and their image coordinates were collected to form a set  $I_c'$ . Also, the real-world coordinates of these nongrid points are measured manually to form another set  $W_c'$ . The set  $I_c'$  then was used to obtain their corresponding real-world coordinates by table lookup using  $T$ , forming a set denoted by  $W_c''$ . Finally, the two sets  $W_c'$  and  $W_c''$  were compared and two types of

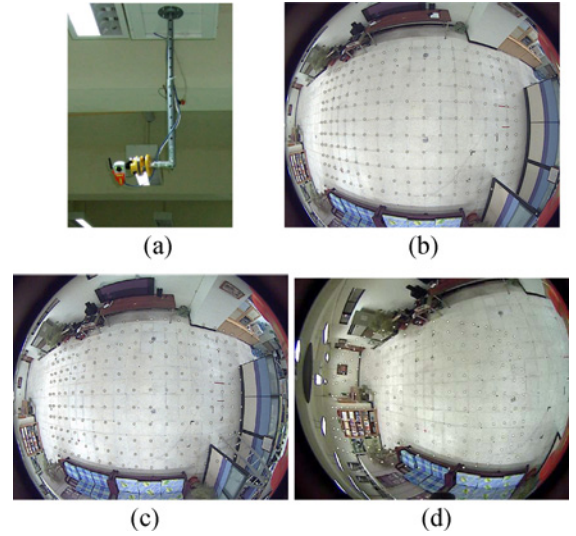


Fig. 6. Fish-eye camera and images used for experiments. (a) Camera setup. (b)–(d) Images taken, respectively, with the camera looking downward at heights 200 and 250 cm, and tilted for  $50^\circ$  at height 200 cm.

error ratio measures were defined to compute the similarity between them.

- 1) *Type-1 location error ratio*  $\varepsilon_1$  with respect to the distance from the real-world point to the camera's lens center:

$$\varepsilon_1 = \frac{\sqrt{(\text{real } x_i - \text{estimated } x_i)^2 + (\text{real } y_i - \text{estimated } y_i)^2}}{\sqrt{\text{real } x_i^2 + \text{real } y_i^2 + L^2}}$$

where *real*  $x_i$  and *real*  $y_i$  are data in  $W_c'$ , and *estimated*  $x_i$  and *estimated*  $y_i$  are data in  $W_c''$ .

- 2) *Type-2 location error ratio*  $\varepsilon_1$  with respect to the effective field of view of the camera as follows:

$$\varepsilon_1 = \frac{\sqrt{(\text{real } x_i - \text{estimated } x_i)^2 + (\text{real } y_i - \text{estimated } y_i)^2}}{\text{radius of effective camera's field of view}}$$

The computed results for the two types of error ratios are summarized in Table I, from which we can see that the ratios are all smaller than 5% which are practical for object location estimation applications like indoor vehicle guidance.

Next, the cameras, still looking downward, were affixed at two different heights 225 and 250 cm, and the corresponding error ratios were computed for some images taken by the camera [an example shown in Fig. 6(c)]. The results were also summarized in Table I, from which we can see the ratios are all smaller than 5% as well. Similarly, for Category-B setups where the cameras were affixed at the height 200 cm and tilted for  $90^\circ$ ,  $70^\circ$ , and  $50^\circ$ , an error ratio table, namely, Table III, was constructed as well for the images taken by the camera [an example shown in Fig. 6(d)]. From the table, we can see that the ratios are *not* all smaller than 5% this time; instead, some are larger (6.4% and 7.5% in the last row in the table). A reason is that the object point dealt with is located at  $(-320, -15)$  which is far away from the image center, falls within a distorted-shaped quadrilateral, and so incurs a larger error in the quadrilateral mapping process. On the other hand, it

TABLE I  
ERROR RATIOS WITH CAMERA LOOKING DOWN AT DIFFERENT CEILING HEIGHTS 200, 225, AND 250 CM

| Real $x, y$<br>(cm)                           | Real distance<br>(cm) | 200 cm                       |   | 225 cm                       |   | 250 cm                       |   |
|---|-----------------------|------------------------------|---|------------------------------|---|------------------------------|---|
|   |                       | Estimated ( $x, y$ )<br>(cm) | Error ratios<br>( <i>type-1, type-2</i> ) | Estimated ( $x, y$ )<br>(cm) | Error ratios<br>( <i>type-1, type-2</i> ) | Estimated ( $x, y$ )<br>(cm) | Error ratios<br>( <i>type-1, type-2</i> ) |
| (-7, -24)                                     | 25                    | (-8, -23)                    | (0.7%, 0.4%)                              | (-8, -24)                    | (0.5%, 0.3%)                              | (-9, -23)                    | (0.9%, 0.7%)                              |
| (-37, 36)                                     | 52                    | (-36, 36)                    | (0.5%, 0.3%)                              | (-37, 35)                    | (0.5%, 0.3%)                              | (-38, 37)                    | (0.6%, 0.4%)                              |
| (-20, 96)                                     | 98                    | (-20, 94)                    | (0.9%, 0.6%)                              | (-20, 95)                    | (0.4%, 0.3%)                              | (-21, 94)                    | (0.8%, 0.7%)                              |
| (-45, -107)                                   | 116                   | (-44, -106)                  | (0.6%, 0.4%)                              | (-46, -107)                  | (0.4%, 0.3%)                              | (-48, -109)                  | (1.3%, 1.1%)                              |
| (-111, -55)                                   | 124                   | (-112, -56)                  | (0.6%, 0.4%)                              | (-114, -56)                  | (1.3%, 1.0%)                              | (-117, -57)                  | (2.3%, 2.0%)                              |
| (-140, 62)                                    | 153                   | (-140, 60)                   | (0.8%, 0.6%)                              | (-143, 61)                   | (1.3%, 1.0%)                              | (-145, 62)                   | (1.7%, 1.6%)                              |
| (-229, -101)                                  | 250                   | (-228, -104)                 | (1.0%, 1.0%)                              | (-233, -106)                 | (2.0%, 2.0%)                              | (-238, -110)                 | (3.6%, 4.0%)                              |
| (-253, 76)                                    | 264                   | (-257, 82)                   | (2.2%, 2.3%)                              | (-260, 82)                   | (2.8%, 2.9%)                              | (-264, 80)                   | (3.2%, 3.7%)                              |
| (-320, -15)                                   | 320                   | (-317, -15)                  | (0.8%, 0.9%)                              | (-331, -14)                  | (2.9%, 3.4%)                              | (-335, -14)                  | (3.9%, 4.7%)                              |
| Average error ratio ( <i>type-1, type-2</i> ) |                       | (0.9%, 0.7%)                 |   | (1.4%, 1.3%)                 |   | (2.0%, 2.1%)                 |   |

TABLE II  
ERROR RATIOS WITH CAMERA AT CEILING HEIGHT 200 CM FOR DIFFERENT TILTED ANGLES 90°, 70°, AND 50°

| Real $x, y$ (cm)                              | Real Distance (cm) | Titled for 90°            |  | Titled for 70°            |  | Titled for 50°            |  |
|---|--------------------|---------------------------|--|---------------------------|--|---------------------------|--|
|   |                    | Estimated ( $x, y$ ) (cm) | Error Ratios ( <i>type-1, type-2</i> ) | Estimated ( $x, y$ ) (cm) | Error Ratios ( <i>type-1, type-2</i> ) | Estimated ( $x, y$ ) (cm) | Error Ratios ( <i>type-1, type-2</i> ) |
| (-7, -24)                                     | 25                 | (-8, -23)                 | (0.7%, 0.4%)                           | (-7, -22)                 | (1.0%, 0.6%)                           | (-7, -22)                 | (1.0%, 0.6%)                           |
| (-37, 36)                                     | 52                 | (-36, 36)                 | (0.5%, 0.3%)                           | (-37, 52)                 | (1.5%, 0.9%)                           | (-38, 34)                 | (1.1%, 0.7%)                           |
| (-20, 96)                                     | 98                 | (-20, 94)                 | (0.9%, 0.6%)                           | (-22, 94)                 | (1.3%, 0.9%)                           | (-21, 90)                 | (2.7%, 1.9%)                           |
| (-45, -107)                                   | 116                | (-44, -106)               | (0.6%, 0.4%)                           | (-45, 103)                | (1.7%, 1.2%)                           | (-42, -103)               | (2.2%, 1.6%)                           |
| (-111, -55)                                   | 124                | (-112, -56)               | (0.6%, 0.4%)                           | (-115, -58)               | (2.1%, 1.6%)                           | (-110, -60)               | (2.2%, 1.6%)                           |
| (-140, 62)                                    | 153                | (-140, 60)                | (0.8%, 0.6%)                           | (-144, 66)                | (2.2%, 1.8%)                           | (-141, 58)                | (1.6%, 1.3%)                           |
| (-229, -101)                                  | 250                | (-228, -104)              | (1.0%, 1.0%)                           | (-224, -95)               | (2.4%, 2.4%)                           | (-238, -110)              | (4.0%, 4.0%)                           |
| (-253, 76)                                    | 264                | (-257, 82)                | (2.2%, 2.3%)                           | (-259, 82)                | (2.6%, 2.6%)                           | (-271, 81)                | (4.5%, 4.6%)                           |
| (-320, -15)                                   | 320                | (-317, -15)               | (0.8%, 0.9%)                           | (-330, -16)               | (2.7%, 3.1%)                           | (-340, -26)               | (6.0%, 7.1%)                           |
| Average error ratio ( <i>type-1, type-2</i> ) |                    | (0.9%, 0.7%)              |  | (1.9%, 1.7%)              |  | (2.8%, 2.6%)              |  |

TABLE III  
COMPARISON OF VEHICLE LOCALIZATION ACCURACY

|                                       | Takeshita <i>et al.</i> | Park <i>et al.</i> | Our Method |
|---------------------------------------|-------------------------|--------------------|------------|
| Average distance error                | 4.6 cm                  | 26 cm              | 5.8 cm     |
| Approximate range of vehicle movement | 2.1 m                   | 7 m                | 6.4 m      |
| Average error percentage              | 5.6%                    | 3.7%               | 0.9%       |

can be seen from the two tables that the estimation accuracy decreases in general as the parameter value of the height, tilt angle, or object point distance increases.

Although the emphasis of this letter is on adaptation of the space-mapping method to camera setup changes, we made additionally a comparison of the location estimation accuracy yielded by our method with those of two other methods [10], [11] whose accuracy results were reported in the literature. The comparison result is shown in Table III, in which the average distance errors, the approximate vehicle movement range, and the average error percentage are shown. Some of the data were computed in this letter from the available data in [10] and [11]. From the table, we can see that the proposed method yields better object location accuracy than the other two methods.

## V. CONCLUSION

A space-mapping method for object location estimation by modifying the basic space-mapping table for camera setup change adaptation has been proposed. The method does not require camera calibration and is general for any camera type. The method estimated object locations by mapping the image coordinates of object points to the real-world ones using a space-mapping table. An algorithm was designed to construct

the table in two stages: construction of a basic table using bilinear interpolation and adaptation of the table to changes of camera heights and orientations, which often occur in different application environments. Experimental results show that the method yields results with error ratios smaller than 5% in most cases, meaning the practicality of the method for various applications.

## REFERENCES

- [1] Z. Yang and W. H. Tsai, "Viewing corridors as right parallelepipeds for vision-based vehicle localization," *IEEE Trans. Ind. Electron.*, vol. 46, no. 3, pp. 653–661, Jun. 1999.
- [2] E. E. Hemayed, "A survey of camera self-calibration," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, Jul. 2003, pp. 351–357.
- [3] O. A. Aider, P. Hoppenot, and E. Colle, "A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences," *Robot. Auton. Syst.*, vol. 52, nos. 2–3, pp. 229–246, Aug. 2005.
- [4] M. Betke and L. Gavruta, "Mobile robot localization using landmarks," *IEEE Trans. Robot. Automat.*, vol. 13, no. 2, pp. 251–263, Apr. 1997.
- [5] H. L. Chou and W. H. Tsai, "A new approach to robot location by house corners," *Patt. Recogn.*, vol. 19, no. 6, pp. 439–451, 1986.
- [6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
- [7] H. C. Chen and W. H. Tsai, "Optimal security patrolling by multiple vision-based autonomous vehicles with omni-monitoring from the ceiling," in *Proc. Int. Comput. Symp.*, Nov. 2008, pp. 196–201.

- [8] S. W. Jeng and W. H. Tsai, "Using pano-mapping tables to unwarping of omni-images into panoramic and perspective-view images," *IET Image Process.*, vol. 1, no. 2, pp. 149–155, Jun. 2007.
- [9] Y. T. Wang and W. H. Tsai, "Indoor security patrolling with intruding person detection and following capabilities by vision-based autonomous vehicle navigation," in *Proc. ICS Workshop Image Process., Comput. Graph., Multimedia Technol.*, Dec. 2006, pp. 995–1000.
- [10] T. Takeshita, T. Tomizawa, and A. Ohya, "A house cleaning robot system: Path indication and position estimation using ceiling camera," in *Proc. Int. Joint Conf. SICE-ICASE*, Oct. 2006, pp. 2653–2656.
- [11] S. Y. Park, S. C. Jung, Y. S. Song, and H. J. Kim, "Mobile robot localization in indoor environment using scale-invariant visual landmarks," in *Proc. IAPR Workshop Cognitive Inform. Process.*, Jun. 2008, pp. 159–163.