Adaptation of Space-Mapping Methods for Object Location Estimation to Camera Setup Changes — A New Study*

Chih-Jen Wu[1] and Wen-Hsiang Tsai[1,2]

[1] Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan
gis91813@cis.nctu.edu.tw
[2] Department of Information Communication, Asia University, Taiwan
whtsai@cis.nctu.edu.tw.

Abstract. A new space-mapping method for object location estimation which is adaptive to camera setup changes in various 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 for establishing a basic table using bilinear interpolation and the second for adapting it to changes of camera heights and orientations. Analytic equations for such adaptation are derived based on image formation and camera geometry properties. Good experimental results are shown to prove the feasibility of the proposed method.

Keywords: object location estimation, space mapping, table adaptation.

1 Introduction

Video cameras are used in various applications, including automatic estimation of the location of an object in an indoor environment using an object image acquired by a camera affixed to a wall or a ceiling. A conventional solution to this problem is to conduct camera calibration to obtain a set of camera parameters, followed by the use the parameters to compute the object location [1-5]. Camera calibration methods often use landmarks to compute camera parameters. The process is generally complicated. An alternative way is to use a *space-mapping table* [6-9] which transforms the image space into the real-world space, thus avoiding camera calibration. The table is constructed with the aid of a *calibration pattern* before the camera is deployed in an application environment.

Space-mapping based 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.

* This work was supported financially by the Ministry of Economic Affairs under Project No. MOEA 97-EC-17-A-02-S1-032 in Technology Development Program for Academia.

J.D. Velásquez et al. (Eds.): KES 2009, Part II, LNAI 5712, pp. 395–402, 2009.

To solve such a problem, one way is to construct a new table in a 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 table construction process. In this study, we investigate the possibility of automatically modifying the original space-mapping table for use in new environments. This is a new topic which has not been studied yet.

In the following, we first describe the idea and the detail of the proposed method in Sections 2 and 3, respectively. Some experimental results are given in Section 4, followed by conclusions in Section 5.

2 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.

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

Stage 1. Construction of a basic mapping table in the factory (see Fig. 1(a)).

Step. 1 Affix the camera to the ceiling at a height H_0 with the camera's optical axis pointing to the floor perpendicularly.

Step. 2 Place a calibration pattern O right under the camera, take an image of it, extract feature points from the image, and find the coordinates of them.

Step. 3 Measure the real-world coordinates of the points in the calibration pattern, which correspond to the extracted feature points in the image.

Step. 4 (*Quadrilateral mapping*) Use a *quadrilateral mapping* technique to construct a *basic space-mapping tabl*e T, which maps each image coordinate pair (u_i, v_j) to a real-world coordinate pair (x_{ij}, y_{ij}), that is, T: $(u_i, v_j) \rightarrow (x_{ij}, y_{ij})$.

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

Step. 5 (*Ceiling-height adaptation*) If the in-field camera setup to be carried out includes just a change of the 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, measure the ceiling height with respective to the floor, and denote it as H_1.

5.2 Modify table T to construct a new one with H_1 as input by a technique of *ceiling height adaptation* described later, 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, measure the ceiling height and the camera's orientation, and denote them as L and θ.

6.2 Modify table T with L and θ as input to be T_1 by a technique of *camera orientation adaptation* described later.

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.

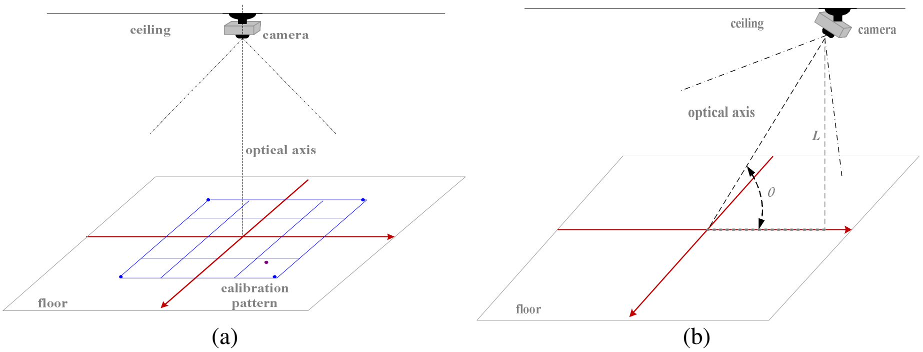

Fig. 1. Illustration of camera setup. (a) Construction of space-mapping table in Stage 1 of proposed method. (b) Camera orientation change with a tilt angle of θ.

3 Basic Space-Mapping Table Construction and Modifications

The quadrilateral mapping technique mentioned in Step 4 of Algorithm 1 as proposed in this study constructs a space-mapping table T by two steps: finding pairs of corresponding quadrilaterals in the calibration pattern in the image and in the real world, followed by transformations of the image and real-world coordinates of corresponding points within the quadrilaterals based on bilinear interpolation, 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* 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 both the floor F_0 at P_0 and the 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, 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 following equalities according to the concept of side proportionality in a triangle:

$$x_1 = x_0 \frac{H_1}{H_0}; \qquad y_1 = y_0 \frac{H_1}{H_0}. \tag{1}$$

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 be magnified in proportion to H_1/H_0 to be (x_1, y_1) as the desired result. Note that we assume here the real-world coordinate system x-y-z is set up at the camera's lens center and the optical axis as the z-axis, and that 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 with a tilt angle of θ 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

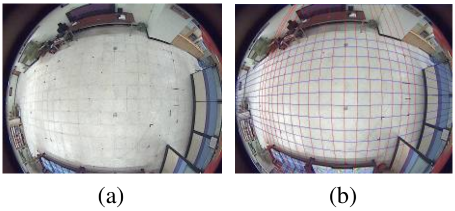

Fig. 2. Illustration of quadrilateral extraction using a grid pattern on floor. (a) An image of the grid pattern. (b) The lines approximating the grid lines.

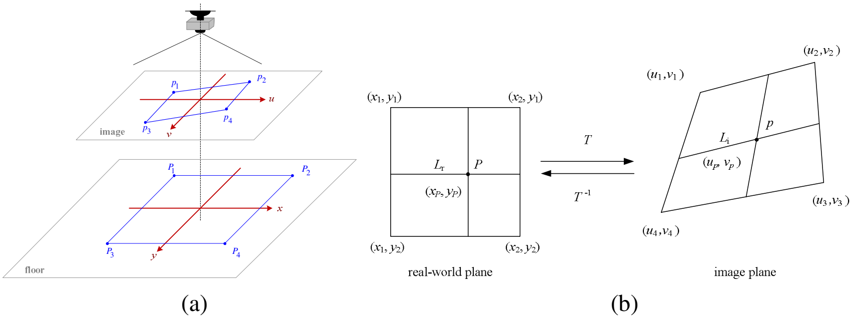

Fig. 3. Quadrilateral mapping. (a) Mapping of corresponding quadrilaterals in image and in calibration pattern. (b) Location estimation of a space point by inverse bilinear interpolation.

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 result, 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 $90^{\circ} - \theta$ 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

$$\frac{x_0}{x'} = \frac{H_0}{H_1}. \tag{2}$$

Also, by geometry and trigonometry we have

$$\sin\theta = \frac{x'}{M}; \tag{3}$$

$$\sin\theta = \frac{L}{N + H_0}; \tag{4}$$

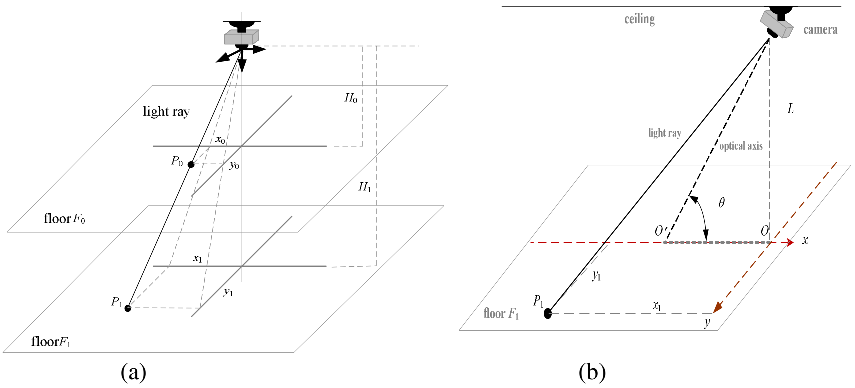

Fig. 4. (a) Use of side proportionality to compute coordinates of point P_1 on a floor F_1 with a ceiling height H_1. (b) A tilted camera with angle θ with respect to the x-axis of the real-world coordinate system.

$$\cos\theta = \frac{x_1 - M}{N + H_0}\,; \tag{5}$$

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

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

$$(x_1 - M)\sin\theta = L\cos\theta. \tag{7}$$

Also, from (2) and (3), we get $x_0 M/\sin\theta = H_0/H_1$, or equivalently,

$$H_1 = \frac{H_0 \sin\theta}{x_0 M}\,. \tag{8}$$

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

$$M = \frac{L}{\sin\theta} \times \frac{x_0}{H_0 \sin\theta - x_0 \cos\theta}\,. \tag{9}$$

And from (7) and (9), we get

$$x_1 = L \times \frac{H_0 \cos\theta + x_0 \sin\theta}{H_0 \sin\theta - x_0 \cos\theta}\,. \tag{10}$$

On the other hand, 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 Eqs. (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

$$y_1 = L \times \frac{y_0}{H_0 \sin\theta - x_0 \cos\theta}\,. \tag{11}$$

4 Experimental Results

A series of experiments have been conducted to test the precision of the proposed method for object location estimation. The fish-eye camera used in the experiments is

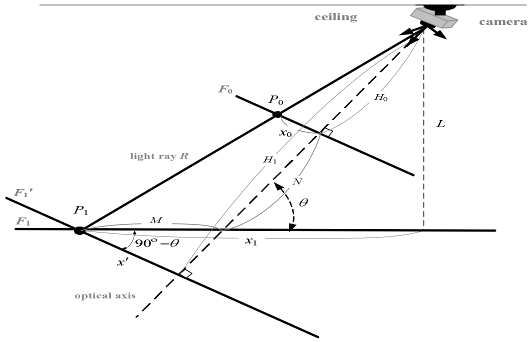

Fig. 5. Lateral view (from the positive *y*-axis direction) of rotation result of floor F_1 in Fig. 4(b) through an angle of $90^o - \theta$ with P_1 as the rotation pivot point

shown in Fig. 6(a), which was attached to a rotator connected to a rod with an adjustable length. The camera can so be tilted arbitrarily and raised to any height. An image taken with the camera looking downward is shown in Fig. 2. We show additionally here three images (Fig. 6(b) through 6(d)) taken with the camera in three distinct setups, which are used in our experiments: (1) looking downward at the height of 200cm; (2) looking downward at the height of 250cm; (3) tilted for the angle of 50^o at the height of 200cm. The images are all of the resolution of 1280×1024.

Case (1) is regarded as the *original* camera setup configuration used for building a basic space-mapping table. After the image of Fig. 6(b) was taken with the downward-looking camera at the height 200cm, all the grid points in the image are extracted to get their image coordinates, forming a set denoted by I_c. Also, the real-world coordinates of each grid point are measured manually to form a set denoted by W_c. The two sets I_c and W_c of coordinate data are then used to construct a basic space-mapping table T by the process described in Section 3. To test the precision of the constructed table T, nine non-grid points among the grid ones, which also appear in Fig. 6(b), were selected and their image coordinates collected to form a set I_c'. Also, the real-world coordinates of these non-grid points are measured manually to form another set W_c'. The set I_c' then is 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'' are compared and two types of error ratio measures are defined to compute the similarity between them: (1) type 1 --- location error ratio with respect to the *distance from the real-world point to the camera's lens center*:

$$\textit{location error ratio} = \frac{\sqrt{(real\ x_i - estimated\ x_i)^2 + (real\ y_i - estimated\ y_i)^2}}{\sqrt{real\ x_i^2 + real\ y_i^2 + L^2}}$$

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* with respect to the *effective field of view of the camera*:

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

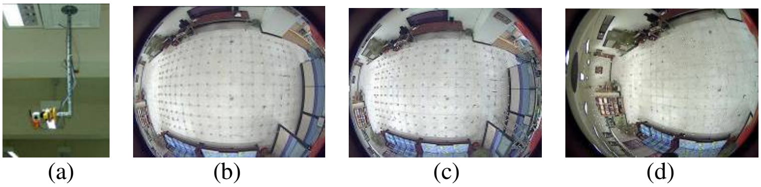

Fig. 6. Fish-eye camera and images used for experiments. (a) The camera can be tilted and lifted arbitrarily. (b)-(d) Images taken with the camera looking downward at heights 200cm and 250cm, and tilted for 50° at height 200cm.

Table 1. Error ratios with camera looking downward at ceiling height 200cm

real x (cm)	*estimated x (cm)*	*real y (cm)*	*estimated y (cm)*	*distance to origin (cm)*	*type-1 error ratio*	*type-2 error ratio*
-7	-8	-24	-23	25	0.7%	0.4%
-37	-36	36	36	52	0.5%	0.3%
-20	-20	96	94	98	0.9%	0.6%
-45	-44	-107	-106	116	0.6%	0.4%
-111	-112	-55	-56	124	0.6%	0.4%
-140	-140	62	60	153	0.8%	0.6%
-229	-228	-101	-104	250	1.0%	1.0%
-253	-257	76	82	264	2.2%	2.3%
-320	-317	-15	-15	320	0.8%	0.9%

Table 2. Error ratios with camera looking downward at ceiling height 250cm

-7	-9	-24	-23	25	0.9%	0.7%
-37	-38	36	37	52	0.6%	0.4%
-20	-21	96	94	98	0.8%	0.7%
-45	-48	-107	-109	116	1.3%	1.1%
-111	-117	-55	-57	124	2.3%	2.0%
-140	-145	62	62	153	1.7%	1.6%
-229	-238	-101	-110	250	3.6%	4.0%
-253	-264	76	80	264	3.2%	3.7%
-320	-335	-15	-14	320	3.9%	4.7%

Table 3. Error ratios with camera tilted for 50° at ceiling height 200cm

real x (cm)	*estimated x (cm)*	*real y (cm)*	*estimated y (cm)*	*distance to origin (cm)*	*type-1 error ratio*	*type-2 error ratio*
-7	-4	-24	-20	25	2.5%	1.6%
-37	-34	36	36	52	1.5%	0.9%
-20	-19	96	93	98	1.4%	1.0%
-45	-39	-107	-103	116	3.1%	2.3%
-111	-106	-55	-59	124	2.7%	2.0%
-140	-138	62	57	153	2.1%	1.7%
-229	-234	-101	-116	250	4.9%	4.9%
-253	-265	76	75	264	3.6%	3.8%
-320	-342	-15	-25	320	6.4%	7.5%

The computed results for the two types of error ratios are summarized as a table shown in Table 1, from which we can see the ratios are all smaller than 5% which is practical for object location estimation applications like indoor vehicle guidance.

For Case (2), the camera, still looking downward, was affixed at a different height 250cm and the previously-mentioned process of error ratio computation was repeated after the proposed method was applied to the image of Fig. 6(c). The results were again summarized as a table shown in Table 2, from which we can see the ratios are all smaller than 5% as well. Similarly, for Case (3) where the camera was affixed at the height 200cm and tilted for 50^o, the error ratio table constructed for the image of Fig. 6(d) is shown in Table 3, from which we see the ratios are *not* all smaller than 5% this time; some are larger (6.4% and 7.5% for the last row in the table). The reason for this phenomenon is that the object point dealt with is located at (−320, −15) which is quite far away from the center of the image, falls within a distorted-shaped quadrilateral, and so incurs a larger error in the quadrilateral mapping process.

5 Conclusions

A general 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 conventional camera calibration, and is general for any camera type. The method estimates the location of an object by mapping the image coordinates of object points to the real-world coordinates of the points using a space-mapping table. An algorithm is designed to construct the table, which consists of two stages, with the first for constructing a basic table using bilinear interpolation and the second for modifying the table to adapt it 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. Chou, H.L., Tsai, W.H.: A new approach to robot location by house corners. Pattern Recognition 19(6), 439–451 (1986)
2. Betke, M., Gurvits, L.: Mobile robot localization using landmarks. IEEE Transactions on Robotics and Automation 13(2), 251–263 (1997)
3. Aider, O.A., Hoppenot, P., Colle, E.: A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. Robotics and Autonomous Systems 52(2-3), 229–246 (2005)
4. Hemayed, E.E.: A survey of camera self-calibration. In: Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, Miami, FL, USA, pp. 351–357 (2003)
5. Yang, Z.F., Tsai, W.H.: Viewing corridors as right parallelepipeds for vision-based vehicle localization. IEEE Trans. Industrial Electronics 46(3), 653–661 (1999)
6. Takeshita, T., Tomizawa, T., Ohya, A.: A house cleaning robot system – path indication and position estimation using ceiling camera. In: Proceedings of International Joint Conference on SICE-ICASE, Busan, Korea, pp. 2653–2656 (2006)
7. Wang, Y.T., Tsai, W.H.: Indoor security patrolling with intruding person detection and following capabilities by vision-based autonomous vehicle navigation. In: Proceedings of 2006 International Computer Symposium (ICS 2006), Taipei, Taiwan (2006)
8. Jeng, S.W., Tsai, W.H.: Using pano-mapping tables to unwarping of omni-images into panoramic and perspective-view Images. IET Image Pro. 1(2), 149–155 (2007)
9. Chen, H.C., Tsai, W.H.: Optimal security patrolling by multiple vision-based autonomous vehicles with omni-monitoring from the ceiling. In: Proceedings of 2008 International Computer Symposium, Taipei, Taiwan (2008)