Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 256896, 16 pages
doi:10.1155/2008/256896

Research Article

Iterative Object Localization Algorithm Using Visual Images with a Reference Coordinate

Kyoung-Su Park,[1] Jinseok Lee,[1] Milutin Stanaćević,[1] Sangjin Hong,[1] and We-Duke Cho[2]

[1] *Mobile Systems Design Laboratory, Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA*
[2] *Department of Electronics Engineering, College of Information Technology, Ajou University, Suwon 443-749, South Korea*

Correspondence should be addressed to We-Duke Cho, chowd@ajou.ac.kr

Received 29 July 2007; Revised 11 March 2008; Accepted 4 July 2008

Recommended by Carlo Regazzoni

We present a simplified algorithm for localizing an object using multiple visual images that are obtained from widely used digital imaging devices. We use a parallel projection model which supports both zooming and panning of the imaging devices. Our proposed algorithm is based on a virtual viewable plane for creating a relationship between an object position and a reference coordinate. The reference point is obtained from a rough estimate which may be obtained from the preestimation process. The algorithm minimizes localization error through the iterative process with relatively low-computational complexity. In addition, nonlinearity distortion of the digital image devices is compensated during the iterative process. Finally, the performances of several scenarios are evaluated and analyzed in both indoor and outdoor environments.

1. INTRODUCTION

The object localization is one of the key operations in many tracking applications such as surveillance, monitoring and tracking [1–8].In these tracking systems, the accuracy of the object localization is very critical and poses a considerable challenge. Most of localization methods use geometric relationship between the object and sensors. Acoustic sensors have been widely used in many localization applications due to their flexibility, low cost, and easy deployment. The acoustic sensor provides directional information in angle of the source with respect to the sensor coordinates which are used to create a geometry for localization. However, an acoustic sensor is extremely sensitive to its surrounding environment with noisy data and does not fully satisfy the requirement of consistent data [9]. Thus as a reliable tracking method, visual sensors are often used for tracking and monitoring systems as well [10, 11]. The visual localization has a potential to yield noninvasive, accurate, and low-cost solution [12–14].

Multiple-image-based multiple-object detection and tracking are used in indoor and outdoor surveillance, and give a delicate and complete history of an interested object's action [2, 15, 16]. The object tracking can be simply concerned into a 2D tracking problem on the ground plane [2, 17–19]. The establishment of correspondences in multiple images can be achieved by using a field of view lines [2, 20]. Besides, for the selection of the best view about interested objects, a camera movement such as zooming and panning is required [19].

There are many localization methods which use image sensors [5, 6, 13, 21–25]. Most of conventional localization methods follow two steps of operation. Initially, the camera parameters are computed offline using known objects or pattern images. Then using additional information such as control points in the scene or techniques such as structure from motion, the relative displacements of a camera are estimated [21, 26]. Basically, these studies can sufficiently localize objects from 3D reconstruction. Once the sufficient number of points is observed in multiple images from different positions, it is mathematically possible to deduce the locations of the points as well as the positions of the original cameras, up to a known factor of scale [21]. In the localization method based on a perspective projection model, the camera calibration is critical. The calibration usually uses a flat plate with a regular pattern [14, 27, 28].

However, in many applications, it is not easy to obtain calibration patterns [29, 30]. In order to alleviate the effect of the calibration patterns, some methods based on self-calibration use the point matching from image sequences [29–34]. In these methods, the image feature extraction should be very accurate since this procedure is very sensitive to the noise [21, 27, 35]. Moreover, if a pair of stereo images for a single scene is not calibrated and the motions between two images are unknown, the image matching requires prohibitively high complexity [27, 34–36].

The localization method based on the affine reconstruction can be used for object localization without the concern of the complex calibration [37–40]. Basically, the relationships between physical space and geometric properties of a set of cameras are considered. The method uses two uncalibrated perspective images where an image is induced by a plane to infinity [37–39, 41–44]. Especially, the factorization method based on the paraperspective projection model can be used for localization [42, 44, 45]. In [42], three well-known approximations such as orthography, weak perspective and paraperspective are involved to full-perspective projection in the affine projection model. In [44, 45], shape and motion recovery is used for less complexity in depth computation. However, the localization method based on the affine structure requires at least five correspondences in two images [37–39]. On the other hand, our proposed method requires only one correspondence (i.e., a centroid coordinate of the detected object) in two images, where each correspondence represents the same object. Thus, the critical requirements of an effective localization algorithm in tracking applications are the computational simplicity with a simpler model where 3D reconstruction is not necessary as well as the robust adaptation of camera's movement during tracking (i.e., zooming and panning) without requiring any additional imaging device calibration from the images. The contribution of this paper is to simplify localization method with efficiency which does not consider 3D reconstruction and complex calibration.

In this paper, we propose a simplified algorithm for localizing multiple objects in a multiple-camera environment, where images are obtained from traditional digital imaging devices. Figure 1 illustrates the application model where multiple people are localized in a multiple-camera environment. The cameras can freely move with zooming and panning capabilities. Within a tracking environment, the proposed method uses detected object points to find object location. We use the 2D global coordinate to represent the object location. In our localization algorithm, the distance between an object and a camera is provided by a reference point. Since the reference point is initially a rough estimate, we are motivated to obtain a more accurate reference point. Here, we use an iterative process which substitutes a previously localized position with a new reference point close to a real-object location. In addition, the proposed localization method has an advantage of using a zooming factor without being concerned about a focal length. Thus, the computational complexity is simplified in determining an object's position which supports both zooming and panning features. In addition, the localization algorithm sufficiently compensates a nonideal property such as optical characteristics of a camera lens.

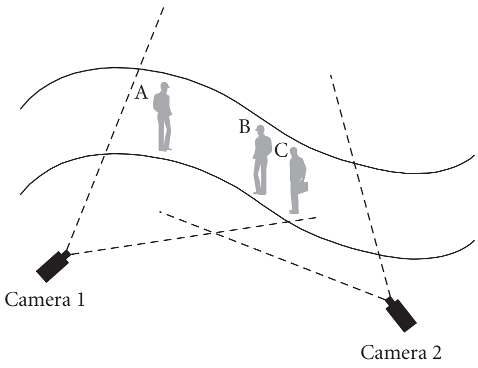


Figure 1: Illustration of the model of application.

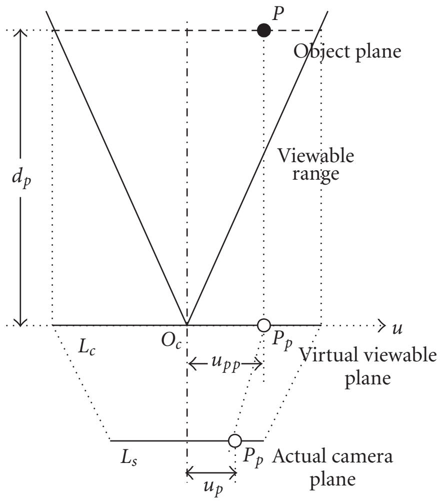


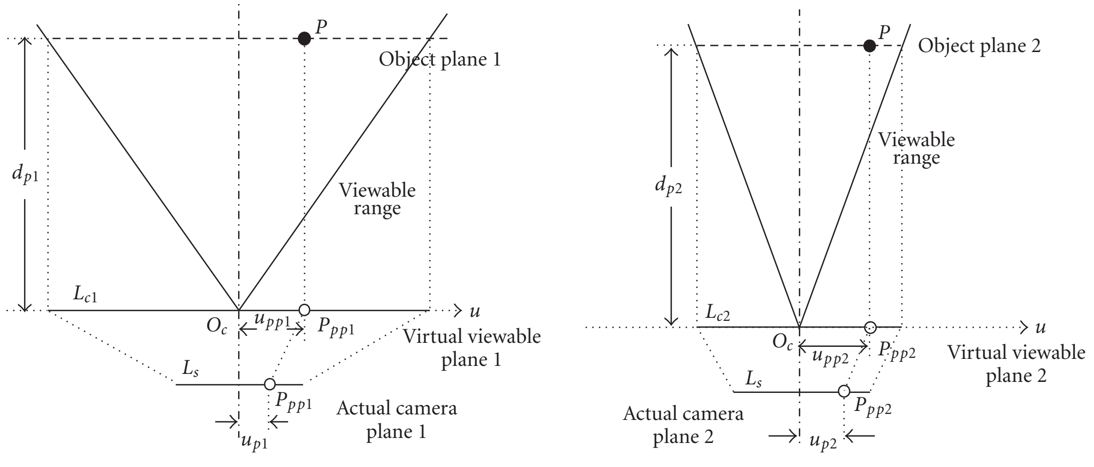
Figure 2: Illustration of the parallel projection model.

The rest of this paper is organized as follows. Section 2 briefly describes a parallel projection model with a single camera. Section 3 illustrates the visual localization algorithm in a 2D coordinate with multiple cameras. In Section 4, we present analysis and simulation results where the localization errors are minimized by compensating for nonlinearity of the digital imaging devices. An application that uses the proposed algorithm for tracking people within a closed environment is illustrated. Section 5 concludes the paper.

2. CHARACTERIZATION OF VIEWABLE IMAGES

2.1. Basic concept of a parallel projection model

In this section, we introduce a parallel projection model to simplify the visual localization, which is basically comprised of three planes: an object plane, a virtual viewable plane and an actual camera plane. In Figure 2, an object P placed on an object plane is projected to both a virtual viewable plane and an actual camera plane, and P_p denotes the projected object point on the virtual viewable plane. The distance d_p denotes the distance between a virtual viewable plane



(a) Small zooming factor (z_1)

(b) Large zooming factor (z_2)

Figure 3: Illustration of the model of zooming in terms of two different zooming factors.

and an object plane. u_p and u_{pp} denote the position of projected object P_p on both the actual camera plane and the virtual viewable plane. The virtual viewable plane is in parallel with the object plane by distance d_p. L_c and L_s denote each length of the virtual viewable plane and the actual camera plane, respectively. The virtual viewable plane is for the connection between the position P on the object plane and the position P_p on the actual camera plane; it has an advantage of simplifying the computation process.

Since the size of image sensor is much smaller than the virtual viewable plane, the viewable range starts from a point O_c. Thus the camera model of parallel projection model is similar to a pin-hole camera. All planes are represented as u- and v-axes but we use u-axis for the explanation of the parallel projection model in this section. Since O_c represents the origin of both the virtual viewable plane and the camera plane, two planes are placed on the same camera position. However, in Figure 2, we drew two planes separately to show the relationship between three planes.

In the parallel projection model, an object is projected from an object plane through a virtual viewable plane to an actual camera plane. Hence, as formulated in (1), u_{pp} is expressed as L_c, L_s, and u_p through the proportional lines of two planes as the following:

$$u_{pp} = u_p\left(\frac{L_c}{L_s}\right). \tag{1}$$

Thus the object P is represented from u_{pp} and the distance d_p between the virtual viewable plane and the object plane.

2.2. Zooming and panning

Since the size of the virtual viewable plane and the object plane are proportional to the distance between the object and the camera (d_p), the length of the virtual viewable plane (L_c) is derived from the distance d_p and the viewable range.

Zooming factor represents the relationship between d_p and L_c. The zooming factor z is defined as a ratio of d_p and L_c as follows:

$$z = \frac{d_p}{L_c}. \tag{2}$$

Since both d_p and L_c use metric units, zooming factor z is a constant.

Figure 3 illustrates the model of zooming in terms of two different zooming factors. Even though the zooming factor of a camera has changed from z_1 to z_2, if the distance between object and camera is not changed, the position of projected object on the virtual viewable plane is not changed. In the figure, since the distance d_{p1} is equal to the distance d_{p2}, the position of the object on the virtual viewable plane is invariant but the position on the actual camera plane is variant. Thus the distance u_{pp1} is equal to u_{pp2} but the distance u_{p1} is different from the distance u_{p2}. The projected positions u_{p1} and u_{p2} on the actual camera planes 1 and 2 are expressed as $u_{pp1} = u_{p1}(L_{c1}/L_s)$ and $u_{pp2} = u_{p2}(L_{c2}/L_s)$. Since $z_1 = d_{p1}/L_{c1}$ and $z_2 = d_{p2}/L_{c2}$, the relationship between u_{p1} and u_{p2} is represented as $u_{p1} = u_{p2}(z_2/z_1)$.

Figure 4 illustrates a special case in which two different objects denoted P_1 and P_2 are projected to the same spot on the actual camera plane. P_{p1} and P_{p2} denote the projected objects on the virtual viewable planes 1 and 2.

The objectsP_1 and P_2 are projected to a point on the actual camera plane while two objects are separated as two different points on the virtual viewable plane 1 and 2. Since the zooming factor z is equal to d_{p1}/L_{c1} and d_{p2}/L_{c2}, the relationship between the distance u_{pp1} and u_{pp2} is expressed as $u_{pp1} = u_{pp2}(d_{p1}/d_{p2})$. The distance u_{p1} is equal to the distance u_{p2}, and the distance u_{pp1} is different from the distance u_{pp2}. It is shown that the distance in projection direction between an object and a camera is an important parameter for the object localization.

Now, we consider a panning factor denoted as θ_p that represents camera rotation. The panning angle is defined as

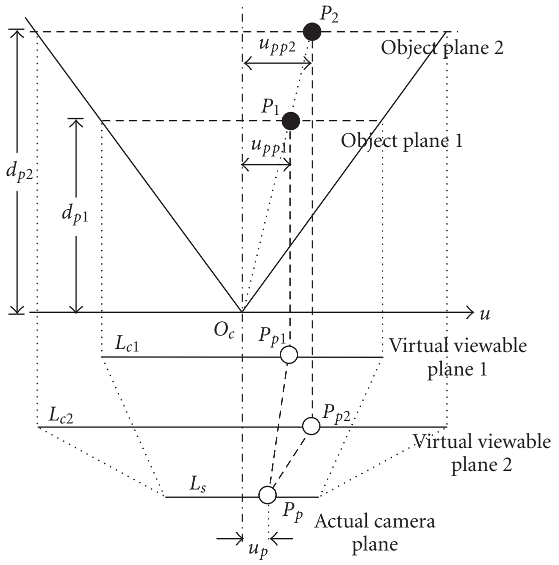


Figure 4: Illustration of a special case in which different objects are projected to the same spot on the actual camera plane.

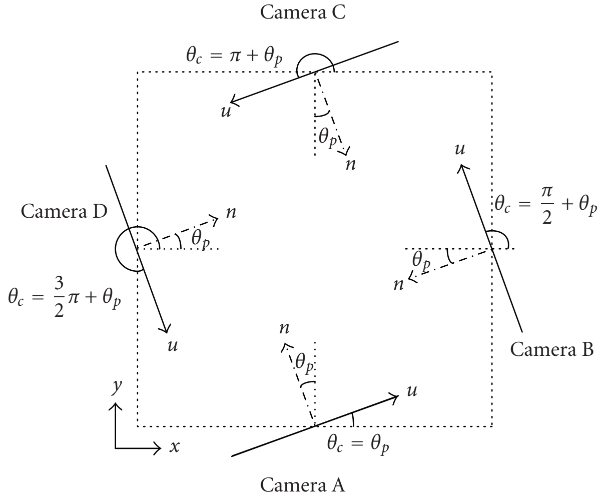


Figure 5: Illustration of individual panning factors with respect to a global coordinate.

the angle difference between n-axis and u-axis where n-axis represents the normal direction of the virtual viewable plane. Thus the panning angle can exist in the range of $-\pi/2 < \theta_p < \pi/2$. The sign of θ_p is determined: the left rotation is positive and the right rotation is negative.

To get the global coordinate of the object, u-axis and v-axis in camera coordinate are translated to x-axis and y-axis in global coordinate. We define camera angle factor (θ_c) to represent the absolute camera angle in global coordinate. The camera angle θ_c is useful to translate the object coordinate from camera images.

Figure 5 illustrates the relationship between the camera angle θ_c and the panning angle θ_p in global coordinate. The global coordinate is represented as x-axis and y-axis. For example, in the position of Camera A, panning angle θ_p is the angle between n- and y-axes; while in Camera D, the panning angle is the angle between n-axis and x-axis. Thus four cases of camera deployment such as Camera A, B, C, D have different relationships between θ_c and θ_p. Thus the projected object $P_p(x_{pp}, y_{pp})$ on the virtual viewable plane is derived from $x_{pp} = x_c + u_{pp}\cos\theta_c$ and $y_{pp} = y_c + u_{pp}\sin\theta_c$. $O_c(x_c, y_c)$ denotes the origin on the virtual viewable plane in global coordinate.

2.3. The relationship between camera positions and pan factors

Figure 6 illustrates the panning factor selection in a pair of cameras depending on an object position. Among deployment of four possible cameras, such as cameras A, B, C, and D, a pair of cameras located in adjacent axes is chosen.

In this paper, we choose cameras$A$ and D for the deployment of two cameras for the sake of the localization formulation. The camera angles in Camera A and D are expressed as $\theta_c = \theta_p$ and $\theta_c = \theta_p + (3/2)\pi$ in terms of the panning angle θ_p.

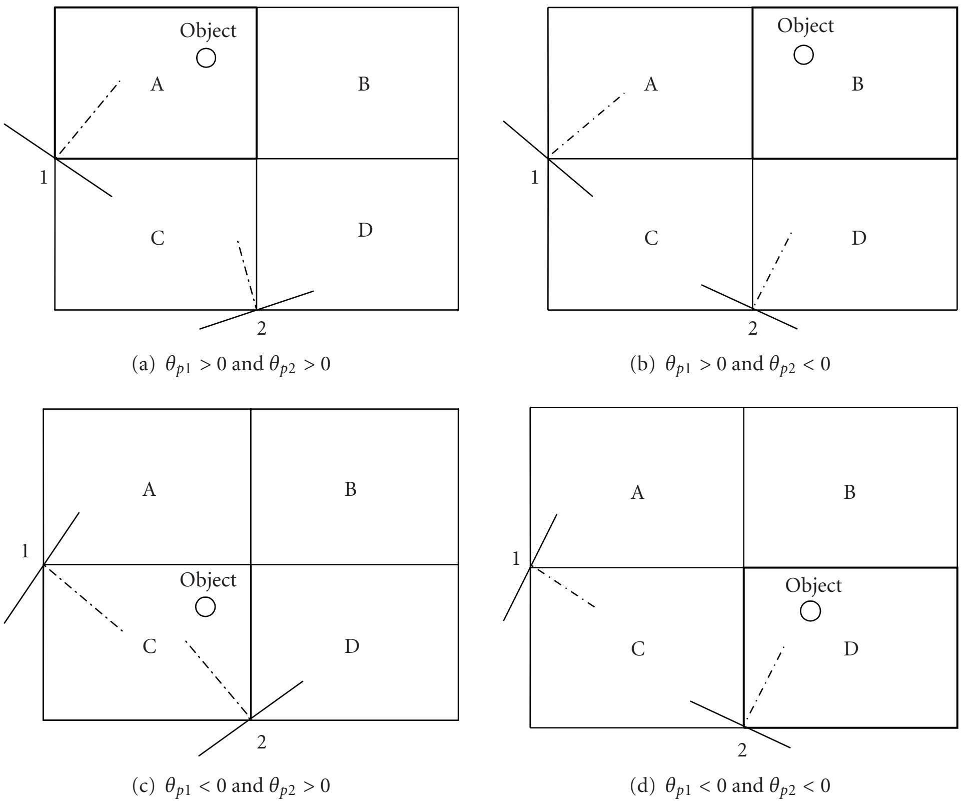
3. VISUAL LOCALIZATION ALGORITHM IN A 2-DIMENSIONAL COORDINATE

3.1. The concept of visual localization

Turning to the object localization with an estimate, consider a single-camera-based localization. In the single-camera localization, we use the estimate plane as an object plane. Figure 7 illustrates the object localization using the estimate E based on a single camera, where E denotes the estimate which is used for a reference point. Note that the the estimate E as a reference point may be any position at the first time, and it becomes close to a real position. The estimate E and the object P are projected to two planes: virtual viewable plane and actual camera plane. Here, the reference point E generates the object plane. The distance d_e denotes the distance between the estimate and the virtual viewable plane. In view of the projected positions, the length l_p is obtained by the length l_{ps}. Hence the object $P(x_p, y_p)$ is determined from the estimate $E(x_e, y_e)$.

Once we use the estimate plane as an object plane, the estimated object position P is different from the real-object position P_r. In other words, since any points on the ray between the object and origin are projected to the same spot on the actual camera plane, the real object P_r is distorted to the point P. Thus, the localization has an error from the distance difference of the distancesd_p and d_e. Through the single-image sensor-based visual projection method, it is shown that an approximated localization is accomplished with a reference point.

We are now motivated to use multiple image sensors in order to reduce the error between P_r and P. In the case of single camera, the distance difference between the distancesd_p and d_e cannot be found by a single-camera view. However, if an additional camera is available for localizing the object within different angles, the distance difference can



(a) $\theta_{p1} > 0$ and $\theta_{p2} > 0$

(b) $\theta_{p1} > 0$ and $\theta_{p2} < 0$

(c) $\theta_{p1} < 0$ and $\theta_{p2} > 0$

(d) $\theta_{p1} < 0$ and $\theta_{p2} < 0$

FIGURE 6: Illustration of panning factor selection in a pair of cameras depending on an object position.

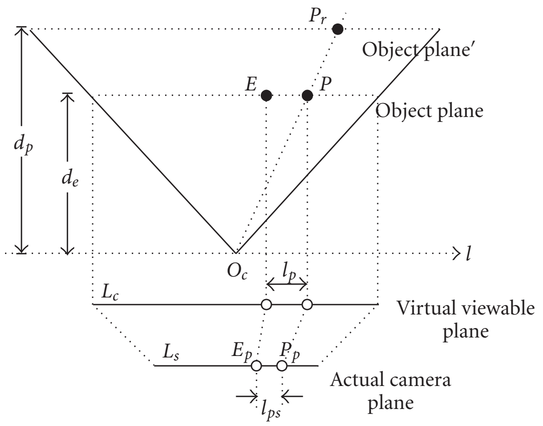


FIGURE 7: Illustration of the visual localization in a single camera.

be compensated by the relationship between two camera views.

Figure 8 illustrates the localization using two cameras for a simple case where both panning factors are zero, and the directions of l_1- and l_2-axes are aligned to y- and x-axes. Given by a reference point E, the virtual viewable planes for two cameras are determined. P_{r1} and P_{r2} are the obtained object coordinates in each single camera. In view of camera 1, the length l_{p1} between the projected points P_{p1} and E_{p1} supports the distance between the object plane of camera 2 and the point P. Similarly, in the view of camera 2, the length l_{p2} between the projected points P_{p2} and E_{p2} supports a distance between the object plane of camera 2 and the point P. Therefore, the basic compensation algorithm is that camera 1 compensates y-direction by the length l_{p1}, and camera 2 compensates x-direction by the length l_{p1} given by a reference point E.

Through one additional image sensor, both l_1 in y-direction and l_2 in x-direction make a reference point $E(x_e, y_e)$ closer to a real-object position. Hence $P(x_p, y_p)$ is computed by $x_p = x_e + l_{p2}$ and $y_p = y_e + l_{p1}$. Note that P is the localized object position through the two cameras, which still results in an error with the real-object position P_r. The error can be reduced by obtaining a reference point E closer to a real position P_r. In Section 3.5, an iterative approach is introduced for improving localization. In the next section, we formulate the multiple image sensor-based localization.

3.2. 2D localization

3.2.1. 2D localization model

In this section, we introduce a simplified localization model. If the estimate E and the object P have the same z-coordinate and v-axis is aligned with z-axis, all points are placed on a plane. Thus the localization is simplified in 2D coordinate. The 2D localization is simple and has an advantage for mapping the test environment. Moreover, once the object

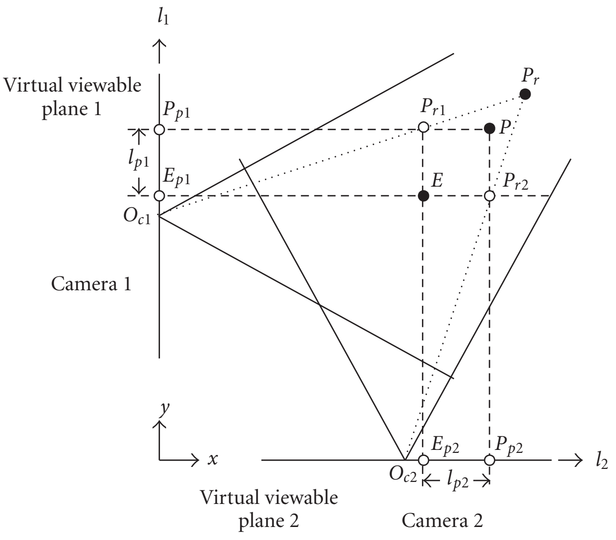


FIGURE 8: Illustration of the localization in multiple cameras.

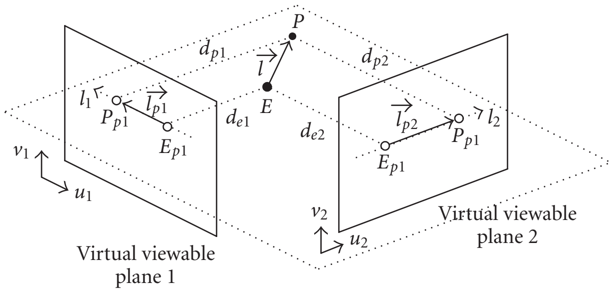
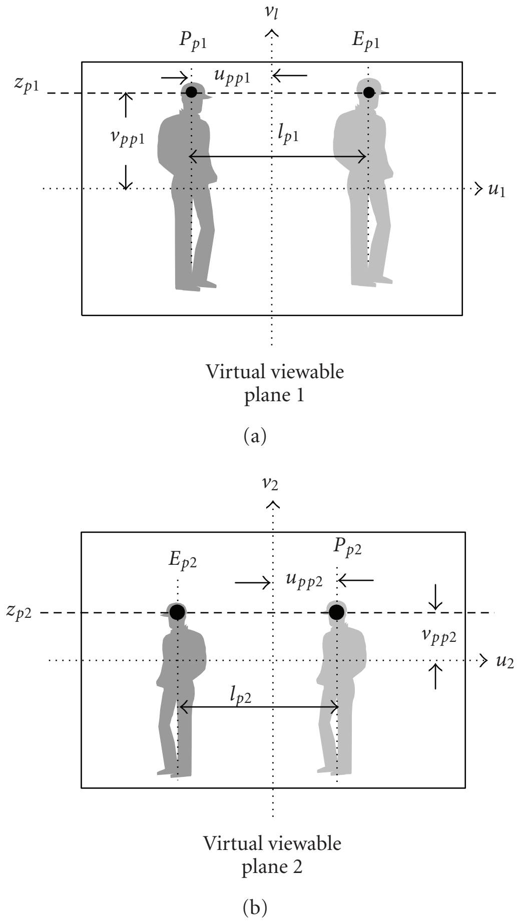


FIGURE 9: Illustration of basic localization algorithm.



(a)

(b)

FIGURE 10: Illustration of the projected images on the virtual viewable planes 1 and 2.

is represented as $P(x_p, y_p)$ in global coordinate, the 2D localization gives a feasible solution.

To derive 2D localization equations, we use vector notation which has a benefit to express the relationship between the estimate and the object where "^" denotes a unit vector and "→" represents a vector. For example, one vector $\vec{r}$ is represented as $A\hat{a}_x + B\hat{a}_y + C\hat{a}_z$, where $\hat{a}_x$, $\hat{a}_y$, and $\hat{a}_z$ denote unit vectors toward x-, y-, and z-axes and A, B, and C are the magnitude of x-, y-, and z-axes, respectively. Figure 9 shows the basic model of object localization. The vectors $\vec{l}$, $\vec{l_{p1}}$, and $\vec{l_{p2}}$ denote the vector from the estimate E to the object P, the vector from the projected estimate E_{p1} to the projected object P_{p1} on the virtual viewable plane 1, and the vector from the projected estimate E_{p2} to the projected object P_{p2} on the virtual viewable plane 2, respectively. The lengthsl_{p1} and l_{p1} are the projections of the vector $\vec{l}$ on the virtual viewable planes 1 and 2.

Figure 10 shows the projected image on the virtual viewable planes 1 and 2 where the projected points P_{p1} and P_{p2} are expressed as P_{p1} (u_{pp1}, v_{pp1}) and P_{p2} (u_{pp2}, v_{pp2}) on the virtual viewable planes 1 and 2. z_{p1} and z_{p2} denote the z-coordinates of the projected objects in global coordinate and are equal to $z_{c1} + v_{pp1}$ and $z_{c2} + v_{pp2}$. Since the estimate has some height with the object, the projected estimate and object have the same z-coordinate on the virtual viewable plane 1 and 2. Thus in the figure, v_{pp1} is different from v_{pp2} while z_{p1} is equal to z_{p2}. Since an estimate is a reference point, the actual estimates in the figure are not displayed on the actual camera plane. Since the projected vectors $\vec{l_{p1}}$ and $\vec{l_{p2}}$ are the projection of vector $\vec{l}$ toward l_1-axis and l_2-axis, the lengths l_{p1} and l_{p2} are equal to $\vec{l}\cdot\hat{a}_{l_1}$ and $\vec{l}\cdot\hat{a}_{l_2}$.

3.2.2. Object localization based on a single camera

The projected object $P_p(l_{pp})$ in l-axis is transformed into $P_p(x_{pp}, y_{pp})$ in global coordinate. The origin $O_c(x_c, y_c)$ is the center of virtual viewable plane. The camera deployment is expressed as the origin $O_c(x_c, y_c)$ and camera angle θ_c.

Figure 11 shows the estimation with a reference point, and a projected object. $\vec{l_p}$ denotes the vector from the origin O_c to the estimate E. The object P, estimate E, projected objects P_p, and projected estimates E_p are denoted as $P(x_p, y_p)$, $E(x_e, y_e)$, $P_p(x_{pp}, y_{pp})$, and $E_p(x_{pe}, y_{pe})$ in global coordinate. The vector $\vec{l_p}$ is expressed in two ways which have different points of view: $\vec{l_p} = (l_{pp} - l_{pe})\hat{a}_l$ on the virtual viewable plane and $\vec{l_p} = (x_p - x_c)\hat{a}_x + (y_p - y_c)\hat{a}_y$ in global coordinate.

The unit vector $\hat{a}_l$ is represented in global coordinate as $\hat{a}_l = \cos\theta_c\hat{a}_x + \sin\theta_c\hat{a}_y$. The vector $\vec{e}$ is expressed as $(x_e - x_c)\hat{a}_x + (y_e - y_c)\hat{a}_y$. Since the length l_{pe} is equal to the projection of vector $\vec{e}$ toward l-axis ($\vec{e}\cdot a_l$), the length l_{pe} is represented as:

$$l_{pe} = (x_e - x_c)\cos\theta_c + (y_e - y_c)\sin\theta_c. \tag{3}$$

Once we assume the estimate is close to the object, the length l_{pp} is represented as

$$l_{pp} = l_{ps}\left(\frac{d_p}{zL_s}\right) \simeq l_{ps}\left(\frac{d_e}{zL_s}\right), \tag{4}$$

where the length l_{ps} is the length of the projected estimate and object on the actual camera plane.

In Figure 11, since the vector $\vec{l_p}$ is equal to $l_{pp}\hat{a}_l - l_{pe}\hat{a}_l$, the length of vector l_p is represented as follows:

$$l_p = l_{pp} - l_{pe}. \tag{5}$$

Since the length l_p is the projection of the vector $\vec{l}$ toward l-axis ($\vec{l}\cdot\hat{a}_l$), the global coordinate $P(x_p, y_p)$ is related with $E(x_e, y_e)$ as follows:

$$(x_p - x_e)\cos\theta_{c1} + (y_p - y_e)\sin\theta_c = l_p. \tag{6}$$

Note that since there are two unknown values of $P(x_p, y_p)$, two equations are necessary.

3.2.3. Object localization based on multiple cameras

As shown in Figure 9, once there are two available cameras which show an object at the same time, two cameras have the following relationship:

$$\begin{aligned}(x_p - x_e)\cos\theta_{c1} + (y_p - y_e)\sin\theta_{c1} &= l_{p1},\\ (x_p - x_e)\cos\theta_{c2} + (y_p - y_e)\sin\theta_{c2} &= l_{p2}.\end{aligned} \tag{7}$$

The projected vector sizes of the vectors $\vec{l_{p1}}$ and $\vec{l_{p2}}$ are derived from $l_{p1} = l_{pp1} - l_{pe1}$ and $l_{p2} = l_{pp2} - l_{pe2}$ in (5). The lengthsl_{pp1} and l_{pp2} are represented as $l_{pp1} \simeq l_{p1}(d_{e1}/z_1L_{s1})$ and $l_{pp2} \simeq l_{p2}(d_{e2}/z_2L_{s2})$ in (4). The length between O_{c1} and P_{p1} in an actual camera plane (l_{p1}) and the length between O_{c2} and P_{p2} in an actual camera plane (l_{p2}) are obtained from displayed images.

Therefore, the object position $P(x_p, y_p)$ is represented as follows:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} \cos\theta_{c1} & \sin\theta_{c1} \\ \cos\theta_{c2} & \sin\theta_{c2} \end{bmatrix}^{-1} \begin{bmatrix} l_{p1} \\ l_{p2} \end{bmatrix}. \tag{8}$$

3.3. Effect of zooming and lens distortion

The errors caused by zooming effect and lens distortion are the reason of scale distortion. In practice, since every general camera lens has nonlinear viewable range, the zooming factor is not a constant. Moreover, since a reference point is a rough estimate, the distance d_p could be different from the distance d_e. However, in (4), the distance d_e, instead of the distance d_p, is used to get the length l_{pp}.

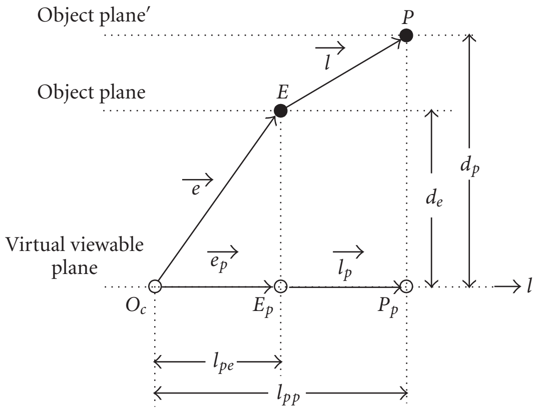


Figure 11: The estimation of a projected object.

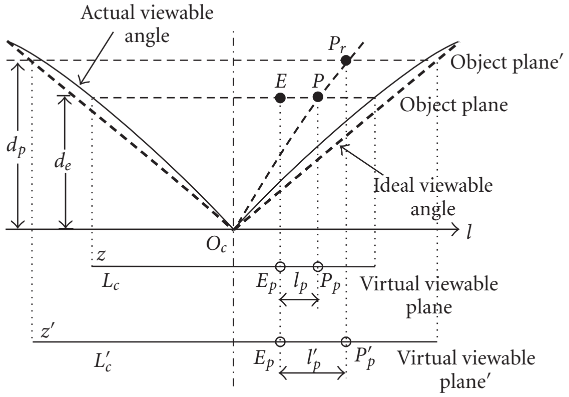


Figure 12: Illustration of actual zooming model caused by lens distortion.

Figure 12 illustrates the actual (nonideal) zooming model caused by lens distortion where the dashed line and the solid line indicate ideal viewable angle and actual viewable angle, respectively.

For reference, zooming distortion is illustrated in Figure 13 with the function of distance from the camera and various actual zooming factors measured by Canon Digital Rebel XT with Tamron SP AF 17–50 mm Zoom Lens [46, 47] where the dashed line is the ideal zooming factor and the solid line is the actual (nonideal) zooming factor. As the distance increases, the nonlinearity property of zooming factor decreases.

To reduce the localization error, we update the length l_p. The lengths l_{pp} and l'_{pp} are equal to $l_{ps}(L_c/L_s)$ and $l_{ps}(L'_c/L_s)$, respectively. Due to the definition of zooming factor, z and z' are expressed as d_e/L_c and d_p/L'_c. Since the objects$P$ and P_r are projected at the same point on the actual camera plane in Figure 12, P and P_r have the same length l_{ps} on the actual camera plane. Thus the actual length l'_p is represented as follows:

$$l'_p = l_{pp}\left(\frac{d_p}{d_e}\right)\left(\frac{z}{z'}\right) - l_{pe}. \tag{9}$$

The distances d_e and d_p are derived from

$$\begin{aligned} d_e &= \sqrt{\left(x_e - x_{pe}\right)^2 + \left(y_e - y_{pe}\right)^2}, \\ d_p &= \sqrt{\left(x_p - x_{pp}\right)^2 + \left(y_p - y_{pp}\right)^2}, \end{aligned} \tag{10}$$

where x_{pe}, y_{pe}, x_{pp}, and y_{pp}, are equal to $x_c + l_{pe}\cos\theta_c$, $y_c + l_{pe}\sin\theta_c$, $x_c + l_{pp}\cos\theta_c$, and $y_c + l_{pp}\sin\theta_c$, respectively.

Finally, the compensated object position $P(x_{pr}, y_{pr})$ is determined as follows:

$$\begin{bmatrix} x_{pr} \\ y_{pr} \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} \cos\theta_{c1} & \sin\theta_{c1} \\ \cos\theta_{c2} & \sin\theta_{c2} \end{bmatrix}^{-1} \begin{bmatrix} l'_{p1} \\ l'_{p2} \end{bmatrix}, \tag{11}$$

where the lengths l'_{p1} and l'_{p2} are equal to $l_{pp1}(d_{p1}/d_{e1})(z_1/z'_1) - l_{pe1}$ and $l_{pp2}(d_{p2}/d_{e2})(z_2/z'_2) - l_{pe1}$, respectively.

3.4. Effect of lens shape

The virtual viewable plane is a plane, and real camera displays a curved space. Thus, unit distances per pixel in u- and v-axes are nonlinear on the actual camera plane. Figure 14 shows the error caused by lens shape, where the distances d_{p1} and d_{p2} denote two different distances between the estimates and the camera.

Figure 15 illustrate the distribution of unit distance of u- and v-axes on the actual camera plane. The distance between camera and calibration sheet is 35 inches and an unit distance is 1 inch.

The translation of the distance between the estimate and the object needs the compensation for the nonlinearity by camera calibration. In Figure 15(a), the unit distance for u-axis is invariant in v-axis and in Figure 15(b), the unit distance for v-axis is also invariant in u-axis. Hence in Figure 10, the height differences of two different cameras have little effect for the overall localization error.

3.5. Iterative localization for error minimization

Once the virtual viewable plane is defined by the estimate, the localized result has the error caused by the distance difference between the estimate E and the real object P_r. Thus the distance between the object and the estimate is important for reducing the localization error.

The basic concept of iterative approach is to use the previous localized position P as a new reference point E for the localization of object P_r. Thus since the reference point E is closer to a real position P_r, the localized position P is getting closer to a real position P_r.

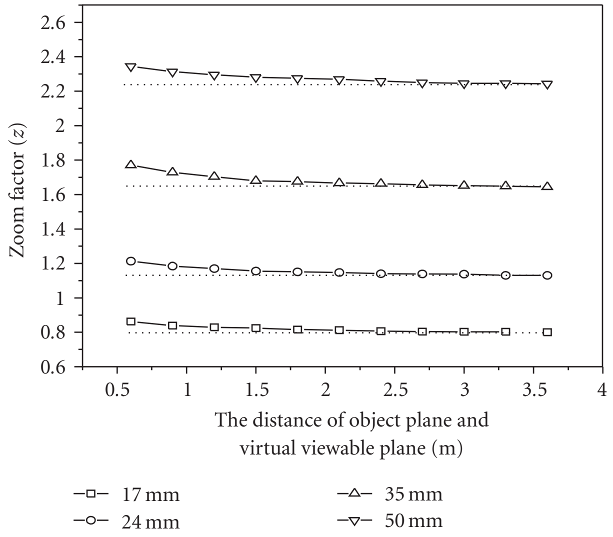


Figure 13: Illustration of zooming distortion on a function of distance from the camera and various actual zooming factors used.

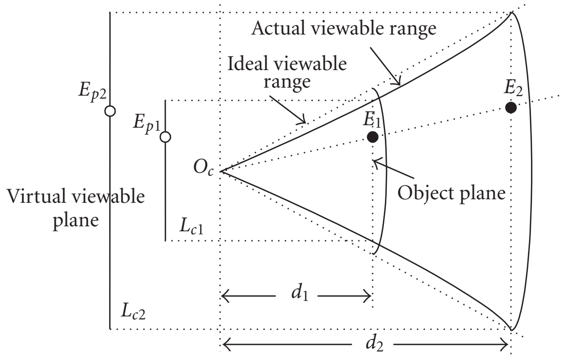


Figure 14: Illustration of the error caused by lens shape.

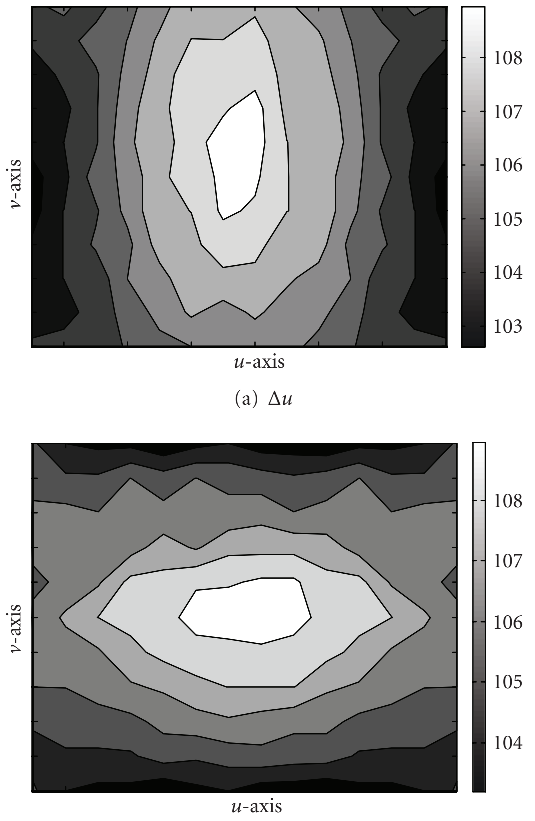
Figure 16(a) illustrates the basic localization based on two cameras where P_r represents the real object. If the distance d_p is equal to the distance d_e, the obtained object coordinate uses the coordinate of P_{p1} and P_{p2} to translate the global coordinate of the object. Thus the object point P is closer to the real object point P_r.

Figure 16(b) shows the iterative localization. Each iteration gives closer object coordinate with relative computational complexity. Thus the iterative approach can reduce the localization error. Furthermore, through the iteration process, the localization is becoming insensitive to the nonlinear properties.

3.6. Effect of tilting angle

In surveillance system, a camera can have tilting angle to increase viewable area. The tilting angle ϕ_c represents the angle difference between z-axis and v-axis on the virtual viewable plane. The tilting angle has the range as $-\pi/2 \leq \phi_c < \pi/2$.

Figure 17 illustrates an example of the tilting angle where one plane is placed on z-axis and the other has θ_c tilting angle. The tilting angle ϕ_c is equal to the angle difference between virtual viewable plane and virtual viewable plane$'$.



(a) Δu

(b) Δv

Figure 15: Illustration of unit distance distribution due to camera nonlinearity on the actual camera plane.

Since u-axis is invariant for the variation of tilting angle, u-axis on the virtual viewable plane is the same as u'-axis on the virtual viewable plane′.

The tilting angle is the reason for distortion in u- and v-axes as shown in Figure 18. $P(u_p, v_p)$ and $P'(u'_p, v'_p)$ denote the project object positions of the same object within different tilting angles. The tilting angle is not affecting the variation in u-axis. However, the tilting angle changes the distance of the object and camera. Thus, once the distance of object and camera is changed, the zooming factor is also changed. Therefore, the tilting angle distorts the object position in u-axis.

In Figure 18, the distance u_p is different from the distance u'_p even if the position of camera and object is not changed. Since u_p and u'_p on the actual camera plane are translated to u_{pp} and u'_{pp} using the zooming factor and the distance between the object and camera, the tilting angle is the reason for the localization error.

Figure 19 illustrates the effect of tilting angle in terms of the distance between the object and the virtual viewable plane. The heightsh_p and h_c denote the object height and the camera height. If the camera has ϕ_c tilting angle, the distance d_p is changed by the distance d'_p.

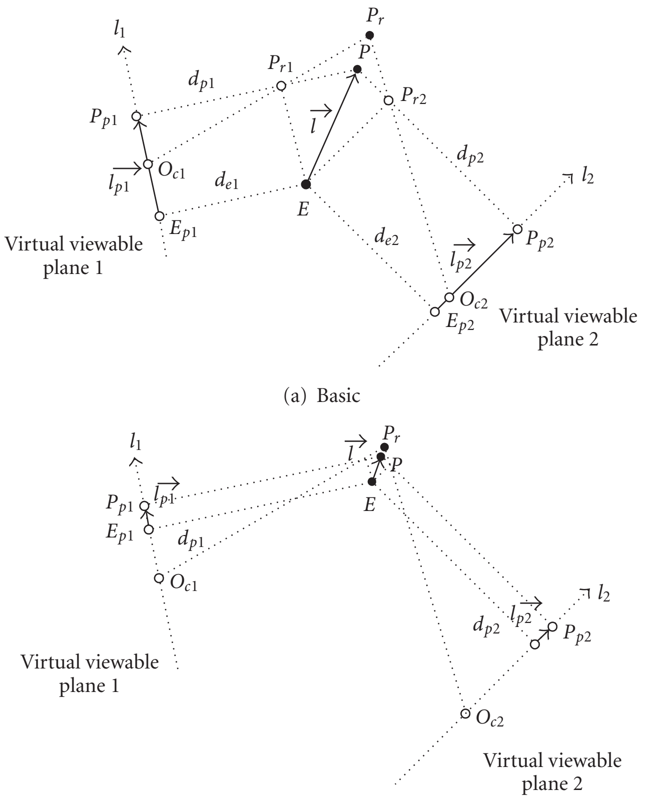
In order to compensate the localization error from tilting angle, we update the distance d_p to d'_p and then change the zooming factor for the distance d'_p. Thus the length l'_p in (9) is updated as follows:

$$l'_p = l_{pp}\left(\frac{d'_p}{d_e}\right)\left(\frac{z}{z'}\right) - l_{pe}, \qquad (12)$$

where z' denotes the zooming factor when the distance between the object and the virtual viewable plane is d'_p.

The distance d'_p is derived as follows:

$$d'_p = (d_p + d_o)\cos\phi_c, \qquad (13)$$



(a) Basic

(b) First iteration

Figure 16: Illustration of iterative localization.

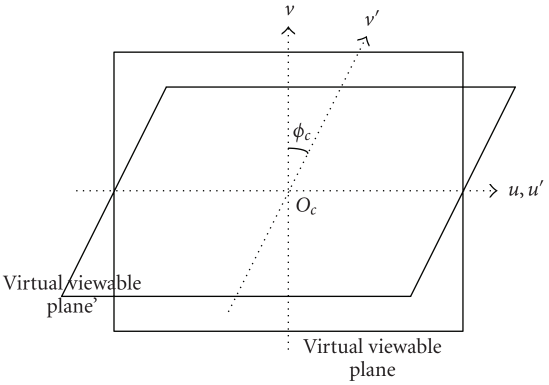
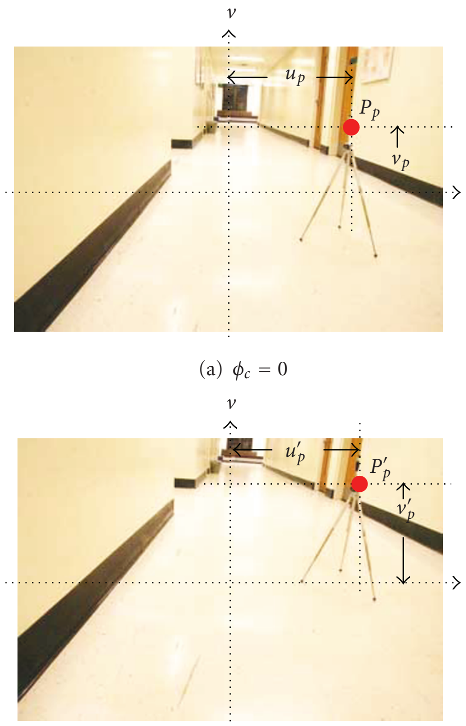


Figure 17: Illustration of an example of the tilting angle.



(a) $\phi_c = 0$

(b) $\phi_c = 10°$

FIGURE 18: Illustration of the distortion by the tilting angle (ϕ_c).

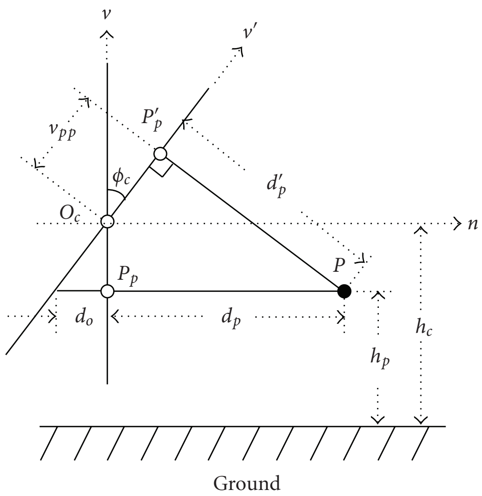
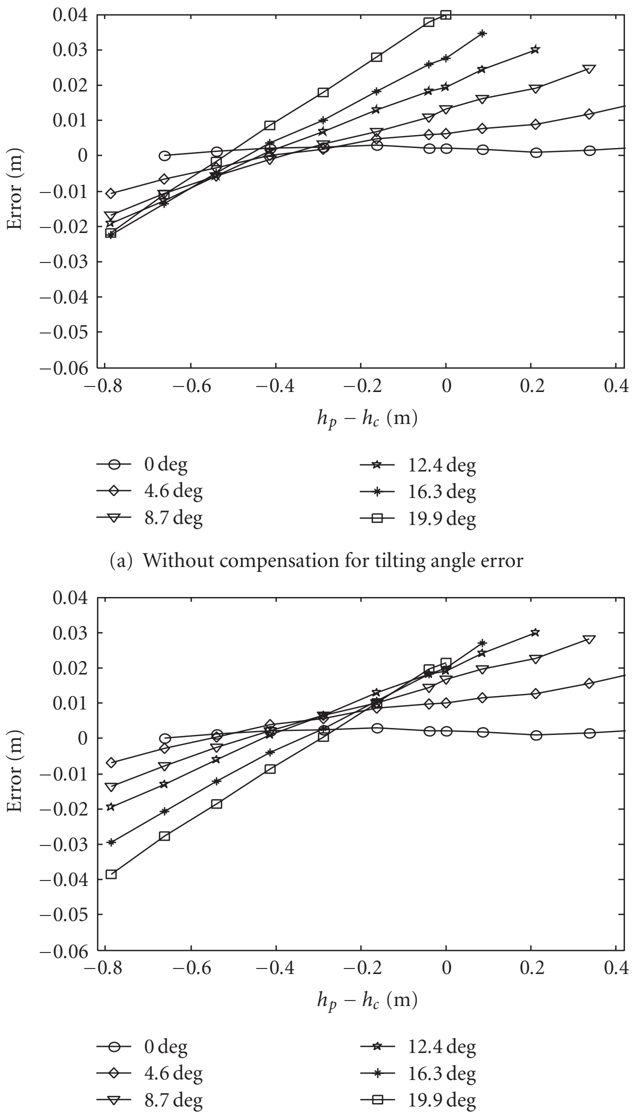


FIGURE 19: Illustration of the effect of tilting angle.

where the distance d_o is computed as

$$d_o = (h_c - h_p)\tan^{-1}\phi_c. \qquad (14)$$

To quantify the localization error caused by tilting angle, we tested the localization error in the simple case. Figure 23 shows the setup of experiment where two cameras are placed on the left side for camera 1 and the bottom side for camera 2 in Cartesian coordinate. For simplicity, the panning factors θ_{p1} and θ_{p2} are both zero. We denote the object is placed on $P(1.8\,\text{m}, 1.8\,\text{m})$ and $E(1.5\,\text{m}, 1.5\,\text{m})$.



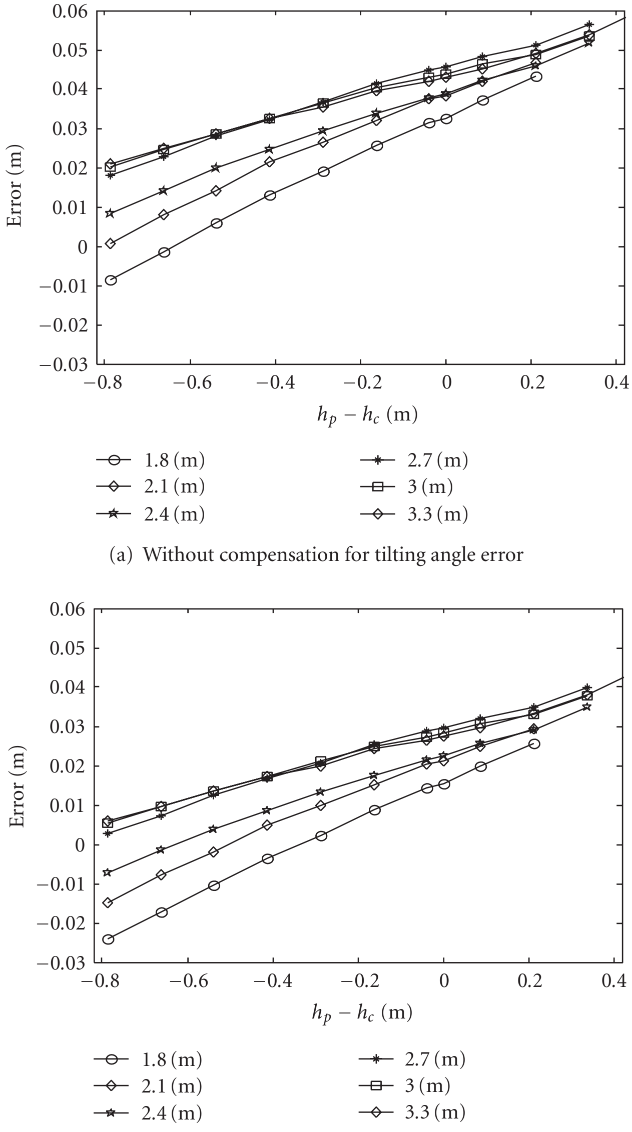
(a) Without compensation for tilting angle error

(b) With compensation for tilting angle error ($h_p - h_c = -0.2$ m)

FIGURE 20: Illustration of the localization error in terms of tilting angle variation.

Figure 20(a) illustrates the localization error in terms of tilting angle variation. If the tilting angle is zero, the height difference between the camera and the object ($h_p - h_c$) does not affect the localization result while the higher tilting angle makes the higher localization error. Thus the tilting angle ϕ_c is the reason for localization error. For example, if the height of the object is 0.2 m lower than the camera height, the range of localization error is from 0.003 to 0.025 m.

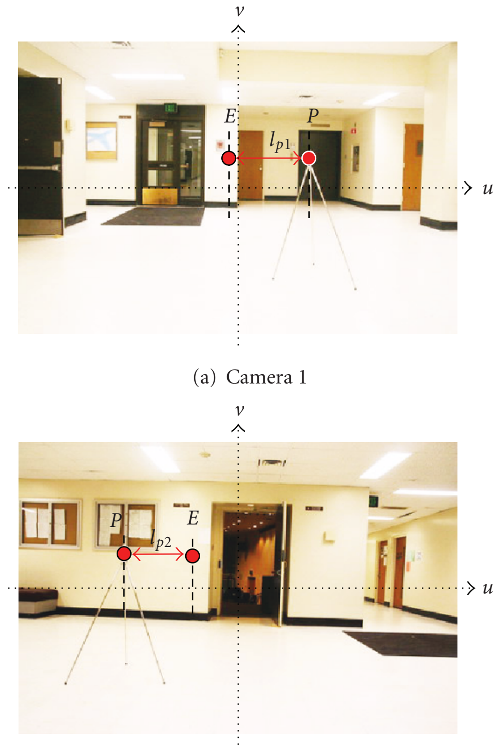
Once object height is provided, the localization error is compensated by (12). In Figure 20(b), we compensated the localization error by denoting the camera height as 1.8 m and the object height as 1.6 m. The overall error caused by the tilting angle has the error range from 0.003 to 0.011 m. If we know the camera height and object height, the error is compensated. Moreover, once the height difference between the object and the camera is unknown, the localization



(a) Without compensation for tilting angle error

(b) With compensation for tilting angle error ($h_p - h_c = -0.2$ m)

Figure 21: Illustration of the localization error in terms of the distance d_p ($\phi_c = 12.4$ deg).



(a) Camera 1

(b) Camera 2

Figure 22: Illustration of two images of camera 1 and camera 2.

error in high-tilting angle, the localization error is obviously improved. Therefore, if we expect the height of the object, the localization error can be successfully compensated.

When the height difference between the object and camera is an unknown value, the compensation for localization caused by tilting angle is difficult. However, if the distance d_p is much longer than the distance d_o, the tilting angle has little effect for the localization error. Figure 21 illustrates the localization error in terms of the distance d_p where the tilting angle is 12.4 degree. When the distance d_p increases, the localization error increases but after d_p is 2.7 m, the error is saturated. In the worst case, the error rate is 0.01 m error per 0.2 m height distance. For example, once the camera height difference is 6 m, the expected error is about 0.3 m. Moreover, when the camera height is 0.2 m taller than the object, the error range is from 0.023 to 0.04 m. Once we assume the object is placed on 0.2 m lower than the camera, the compensation reduces the error to the range of 0.006 to 0.024 m.

4. ANALYSIS AND SIMULATION

4.1. Simulation setup: basic illustration

The objective in this simulation ensures the proposed localization algorithm by measuring the localization error in the real case. To show the compensation for camera nonlinearity, we chose small space which is close to the camera. In the case of Figure 13, the distortion from camera nonlinearity exists in 2.0 m inside space. Thus in this simulation, we use 4 m × 4 m area.

Our target application is a surveillance system where most of target objects are human or vehicle. However, in this simulation, we use a small ball as a target object to simplify the target detection. There are many reasons for localization error caused by detection. For example, the centroid detection of a human is important for reducing localization error since a human is represented as a point. If we use different positions between two camera images, the localization result has some centroid error. Thus in this setup, we use a small ball. Moreover, after taking pictures, we manually search the center of ball. We analyze the localization error in 2D global coordinate. The object is represented as $P(x_p, y_p)$.

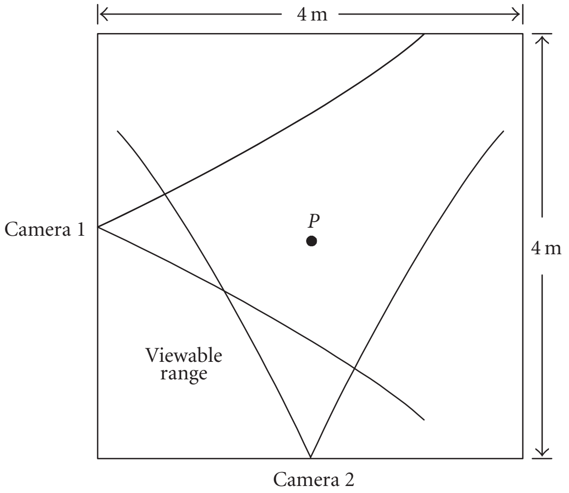
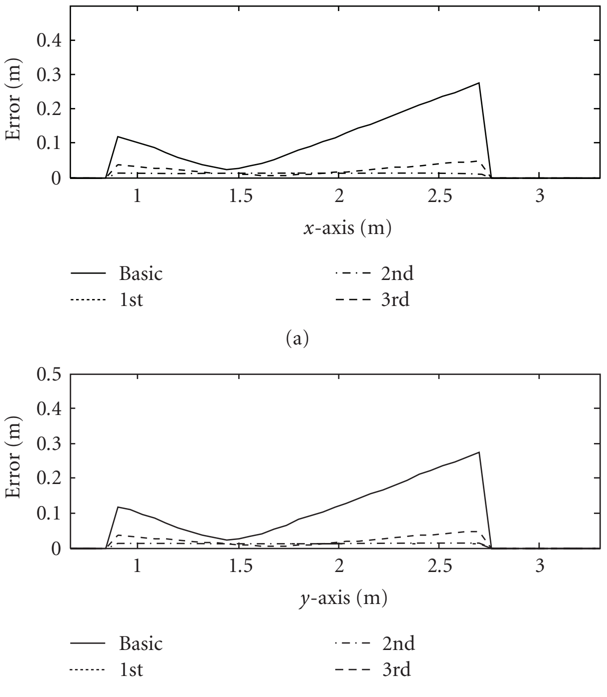


Figure 23: Illustration of experimental setup for localizing an actual object.



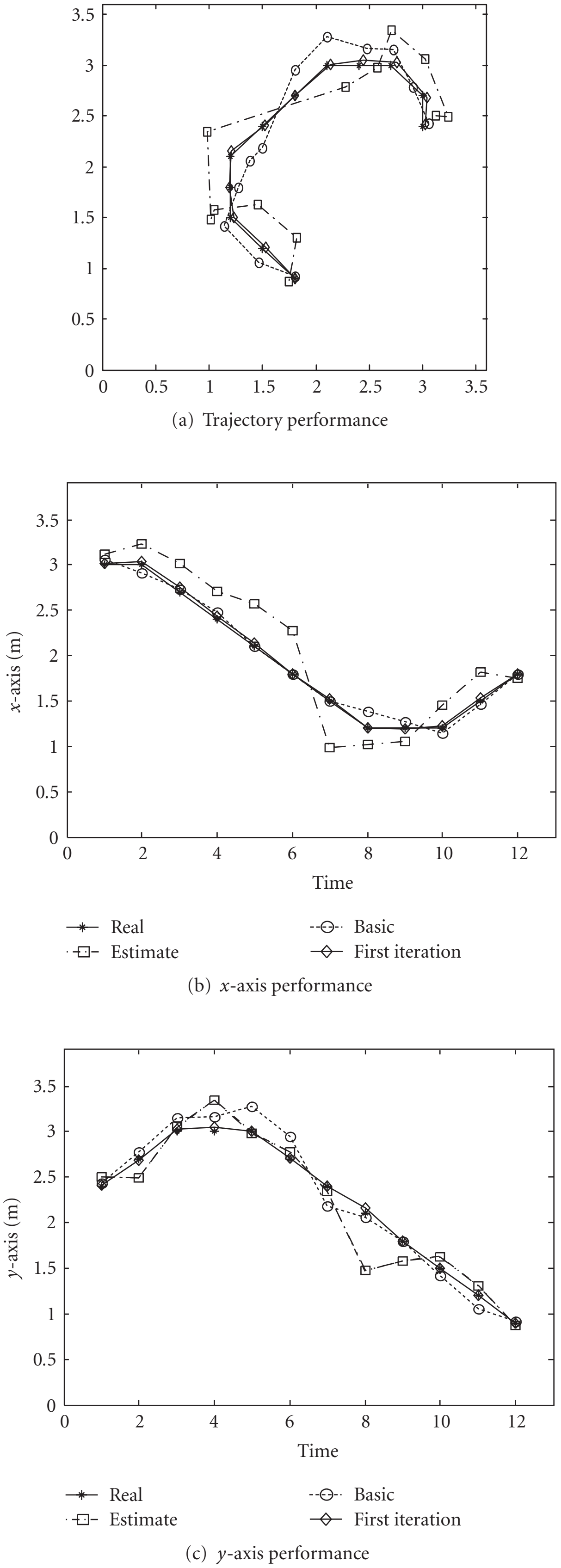
(a)

(b)

Figure 24: Illustration of error comparison based on the number of iterations.

Figure 22 shows the displayed images in two cameras where the lengthsl_{p1} and l_{p2} are distances between a reference point E and a real-object point P in camera 1 and camera 2, respectively. To explain the test setup, we showed the reference point E in Figures 22(a) and 22(b), but actually the reference point is a virtual point.

Figure 23 shows the experiment setup to measure an actual object. In this experiment, the actual position of the object is calculated from the reference based on the parallel projection model. In Figure 23, two cameras are placed on



(a) Trajectory performance

(b) x-axis performance

(c) y-axis performance

Figure 25: Application of the noniterative localization in tracking a trajectory with rough estimates.

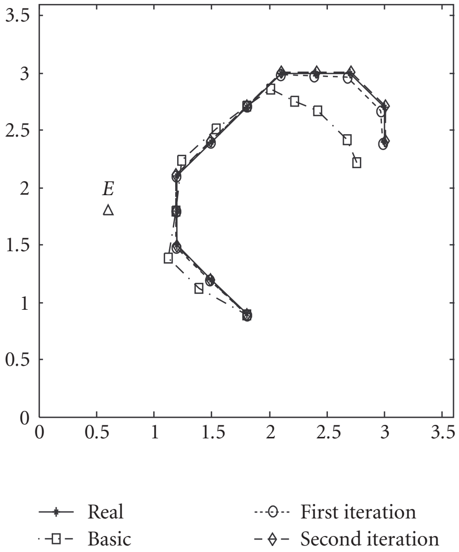
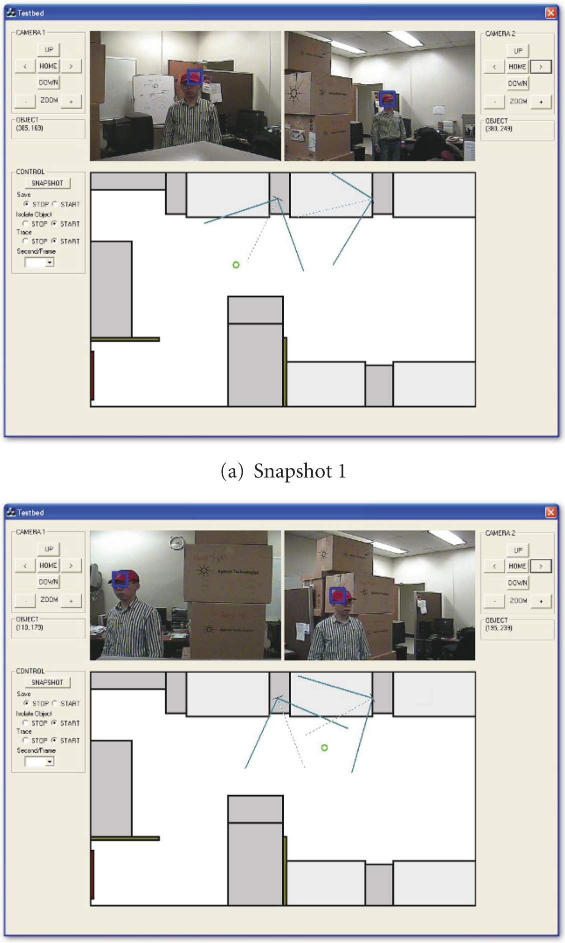


FIGURE 26: Application of the iterative localization with single estimate.



OBJECT (365, 169)

Save ● STOP ○ START

Isolate Object ○ STOP ● START

Trace ○ STOP ● START

Second/Frame

OBJECT (380, 249)

(a) Snapshot 1

OBJECT (110, 179)

Save ● STOP ○ START

Isolate Object ○ STOP ● START

Trace ○ STOP ● START

Second/Frame

OBJECT (195, 239)

(b) Snapshot 2

FIGURE 27: The snapshots of the tracking environment with moving camera based on the proposed localization algorithm. Human face is used to localize a person. The circle represents the actual coordinate of the person within the room.

the left side for camera 1 and the bottom side for camera 2 in Cartesian coordinate. Both camera panning factors θ_{p1} and θ_{p2} are at zero.

The actual zooming factors are $z_1 = d_{e1}/L_{c1}$ and $z_2 = d_{e2}/L_{c2}$, where z_1' is the zooming factor when the distance between the object plane and the virtual viewable plane is d_{p1}, and z_2' is the zooming factor when the distance between the object plane and the virtual viewable plane is d_{p2}. Now, we analyze the localization result and compare the localization error depending on the iteration process called compensation.

4.2. Localization error and object tracking performance

Figure 24 shows the error distribution of the algorithm where two cameras are positioned at $O_{c1}(1.8, 0)$ and $O_{c2}(0, 1.8)$. The actual object is located at $P(1.5, 1.5)$. The figures illustrate the amount of localization error as a function of the reference coordinate. Since each camera has limited viewable angles, the reference coordinate located on the outside of viewable angle cannot be considered. Note that the error is minimized when the reference points are close to the actual object point. The localization error can be further reduced with multiple iterations.

The proposed localization algorithm is also used for a tracking example. In this example, an object moves within a 4 m × 4 m area, and the images are obtained from the real cameras. We first applied the proposed noniterative localization algorithm with compensation in tracking problems. Each time the object changes coordinates, its corresponding estimation is generated. Figure 25(a) illustrates the trajectory result of localization. After the compensation, the tracking performance is improved. Figures 25(b) and 25(c) illustrate the tracking performance in the x-axis and the y-axis. These figures clearly show that the compensation improves the tracking performance but the localization error still exists.

Similarly, the proposed iterative localization algorithm is used in the same tracking example. In this case, only one reference coordinate is used for the entire localization. The chosen estimate is outside the trajectory as shown in Figure 26. This figure illustrates the trajectory result of localization. There is a significant error with the one iteration since the estimated coordinate is not close to the object. Note that the error increases if the object is further away from the estimated coordinate. However, successive iterations eliminated the localization error as shown in the figure.

4.3. Application of the algorithms

Figure 27 shows a tracking environment with moving cameras where the proposed localization algorithm is applied. For illustration, two sequences of images are shown. The

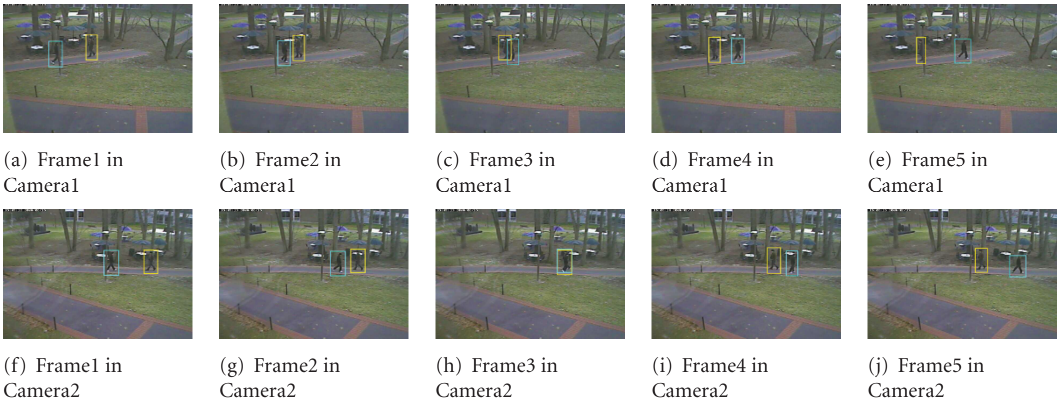


Figure 28: Illustration of detection results for people localization in an outdoor environment.

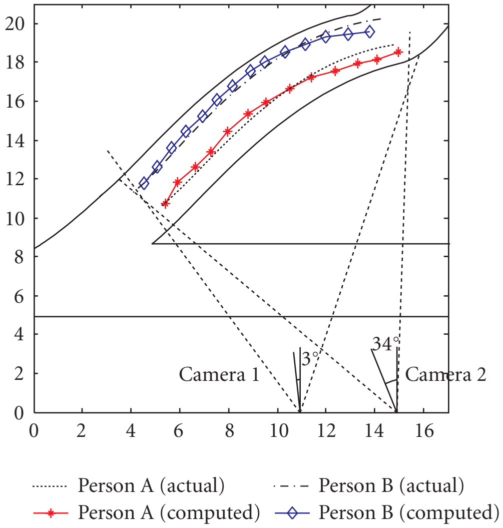


Figure 29: Illustration of two objects trajectory in an outdoor environment.

coordinates of the center of the room is chosen as the initial reference coordinate. The cameras follow the object during the localization. When the object is detected by individual camera, the coordinate of the camera images are combined for actual coordinate. The actual coordinate is shown in the tracking environment. In the experiment, cameras are following the object through panning.

Figure 28 illustrates object detection in outdoor environment where two objects are used for evaluating the proposed localization algorithm. Both cameras are placed on the same side and the panning angles for camera 1 and camera 2 are 3° and 34°, respectively. Figure 29 illustrates two objects trajectories in an outdoor environment. Since the method is computationally simple, the total computation time is proportional to the the number of objects, which is not a significant with respect to overall computation. As shown in the figure, the trajectory computation errors are negligibly small for the practical use. The average error in terms of the distance between the actual trajectories and the computed trajectories is 0.294 m and 0.296 m for personsA and B, respectively. However, the maximum error can go up to as much as 0.608 m (3%) for person B. In addition to the localization algorithm computation errors, note that additional contributing factors on the errors are the measurements of the distances between the cameras and persons, and the selected center point of the detected regions of the persons used in the computation.

5. CONCLUSION

This paper proposes an accurate and effective object localization algorithm with visual images from unreliable estimate coordinates. In order to simplify the modeling of visual localization, the parallel projection model is presented where simple geometry is used in computation. The algorithm minimizes the localization error through iterative approach with relatively low-computational complexity. Nonlinearity distortion of the digital image devices is compensated during the iterative approach. The effectiveness of the proposed algorithm in object position localization as well as tracking is illustrated. The proposed algorithm can be effectively applied in many tracking applications where visual imaging devices are used.

ACKNOWLEDGMENTS

This research is supported by Foundation of Ubiquitous Computing and Networking (UCN) project, the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea, and a result of subproject UCN 08B3-O4-30S.

REFERENCES

[1] R. Okada, Y. Shirai, and J. Miura, “Object tracking based on optical flow and depth,” in *Proceedings of the IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 565–571, Washington, DC, USA, December 1996.

[2] S. Khan and M. Shah, “Consistent labeling of tracked objects in multiple cameras with overlapping fields of view,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355–1360, 2003.

[3] A. Bakhtari, M. D. Naish, M. Eskandari, E. A. Croft, and B. Benhabib, "Active-vision-based multisensor surveillance—an implementation," *IEEE Transactions on Systems, Man, and Cybernetics C*, vol. 36, no. 5, pp. 668–680, 2006.

[4] N. X. Dao, B.-J. You, S.-R. Oh, and Y. J. Choi, "Simple visual self-localization for indoor mobile robots using single video camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, vol. 4, pp. 3767–3772, Sendai, Japan, September 2004.

[5] V. Ayala, J. B. Hayet, F. Lerasle, and M. Devy, "Visual localization of a mobile robot in indoor environments using planar landmarks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, vol. 1, pp. 275–280, Takamatsu, Japan, October 2000.

[6] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, "A joint particle filter for audio-visual speaker tracking," in *Proceedings of the 7th International Conference on Multimodal Interfaces (ICMI '05)*, pp. 61–68, Torento, Italy, October 2005.

[7] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Joint audio-visual tracking using particle filters," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 11, pp. 1154–1164, 2002.

[8] G. Pingali, G. Tunali, and I. Carlbom, "Audio-visual tracking for natural interactivity," in *Proceedings of the 7th ACM International Conference on Multimedia*, pp. 373–382, Orlando, Fla, USA, October 1999.

[9] D. B. Ward, E. A. Lehmann, and R. C. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, 2003.

[10] H. Lee and H. Aghajan, "Collaborative node localization in surveillance networks using opportunistic target observations," in *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, pp. 9–18, Santa Barbara, Calif, USA, October 2006.

[11] O. Yakimenko, I. Kaminer, and W. Lentz, "A three point algorithm for attitude and range determination using vision," in *Proceedings of the American Control Conference (ACC '00)*, vol. 3, pp. 1705–1709, Chicago, Ill, USA, June 2000.

[12] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras," in *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '01)*, pp. 91–96, Baden-Baden, Germany, August 2001.

[13] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[14] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[15] M. Han, A. Sethi, W. Hua, and Y. Gong, "A detection-based multiple object tracking method," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3065–3068, October 2004.

[16] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[17] H. Jin and G. Qian, "Robust multi-camera 3D people tracking with partial occlusion handling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 1, pp. 909–912, Honolulu, Hawaii, USA, April 2007.

[18] J. Berclaz, F. Fleuret, and P. Fua, "Robust people tracking with global trajectory optimization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 744–750, New York, NY, USA, June 2006.

[19] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool, "Color-based object tracking in multi-camera environments," in *Proceedings of 25th DAGM Symposium on Pattern Recognition*, pp. 591–599, Magdeburg, Germany, September 2003.

[20] O. Javed, S. Khan, Z. Rasheed, and M. Shah, "Camera handoff: tracking in multiple uncalibrated stationary cameras," in *Proceedings of the IEEE Workshop on Human Motion (HUMO '00)*, pp. 113–118, Los Alamitos, Calif, USA, December 2000.

[21] P. E. Debevec, *Modeling and rendering architecture from photographs*, Ph.D. thesis, University of California at Berkeley Computer Science Division, Berkeley Calif, USA, 1996.

[22] M. Watannabe and S. K. Nayar, "Telecentric optics for computational vision," in *Proceedings of the 4th European Conference on Computer Vision (ECCV '96)*, vol. 2, pp. 439–451, Cambridge, UK, April 1996.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] C. Geyer and K. Daniilidis, "Omnidirectional video," *The Visual Computer*, vol. 19, no. 6, pp. 405–416, 2003.

[25] S. Spors, R. Rabenstein, and N. Strobel, "A multi-sensor object localization system," in *Proceedings of the Vision Modeling and Visualization Conference (VMV '01)*, pp. 19–26, Stuttgart, Germany, November 2001.

[26] S. Bougnoux, "From projective to Euclidean space under any practical situation, a criticism of self-calibration," in *Proceedings of the 6th IEEE International Conference on Computer Vision (ICCV '98)*, pp. 790–796, Bombay, India, January 1998.

[27] R. K. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 713–720, 1988.

[28] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 1106–1112, San Juan, Puerto Rico, USA, June 1997.

[29] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1513–1518, 2006.

[30] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: theory and experiments," in *Proceedings of the 2nd European Conference on Computer Vision (ECCV '92)*, pp. 321–334, Santa Margherita Ligure, Italy, May 1992.

[31] A. Zisserman, P. A. Beardsley, and I. D. Reid, "Metric calibration of a stereo rig," in *Proceedings of the IEEE Workshop on Representation of Visual Scenes (WVRS '95)*, pp. 93–100, Cambridge, Mass, USA, June 1995.

[32] E. Horster, R. Lienhart, W. Kellermann, and J.-Y. Bouguet, "Calibration of visual sensors and actuators in distributed computing platforms," in *Proceedings of the 3rd ACM International Workshop on Video Surveillance & Sensor Networks*, pp. 19–28, Hilton, Singapore, November 2005.

[33] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: a general algorithm, singularities, applications," in *Proceedings of the IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition (CVPR '99), vol. 1, pp. 432–437, Fort Collins, Colo, USA, June 1999.

[34] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 87–119, 1995.

[35] Q. Memon and S. Khan, "Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks," *International Journal of Systems Science*, vol. 32, no. 9, pp. 1155–1159, 2001.

[36] R. Cipolla, T. W. Drummond, and D. Robertson, "Camera calibration from vanishing points in images of architectural scenes," in *Proceedings of the British Machine Vision Conference*, vol. 2, pp. 382–391, Nottingham, UK, September 1999.

[37] P. A. Beardsley, A. Zisserman, and D. W. Murray, "Sequential updating of projective and affine structure from motion," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 235–259, 1997.

[38] O. Faugeras, "Stratification of three-dimensional vision: projective, affine, and metric representations: errata," *Journal of Optical Society of America*, vol. 12, no. 3, pp. 465–484, 1995.

[39] T. Moons, L. Van Gool, M. Proesmans, and E. Pauwels, "Affine reconstruction from perspective image pairs with a relative object-camera translation in between," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 77–83, 1996.

[40] M. Pollefeys and L. Van Gool, "A stratified approach to metric self-calibration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 407–412, San Juan, Puerto Rico, USA, June 1997.

[41] P. A. Beardsley and A. Zisserman, "Affine calibration of mobile vehicles," in *Proceedings of the Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision (GMICV '95)*, Xi'an, China, April 1995.

[42] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *Journal of the Optical Society of America A*, vol. 8, no. 2, pp. 377–385, 1991.

[43] P. Sturm and L. Quan, "Affine stereo calibration," in *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns (CAIP '95)*, pp. 838–843, Prague, Czech Republic, September 1995.

[44] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[45] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206–218, 1997.

[46] http://www.usa.canon.com/.

[47] http://www.tamron.com/.