TOUR GUIDANCE BY CAR DRIVING IN PARK AREAS USING AUGMENTED REALITY AND OMNI-VISION TECHNIQUES

[1] *Bo-Cheng Chen* (陳柏丞) and [2]*Wen-Hsiang Tsai* (蔡文祥)

[1]Institute of Multimedia Engineering
[2]Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
Emails: ae123481@hotmail.com, whtsai@cis.nctu.edu.tw

ABSTRACT

An augmented-reality based tour guidance system for use in park areas using a video surveillance vehicle and computer vision techniques has been proposed. When a user drives a vehicle in a park, by the proposed system he/she can get guidance information about the names of the buildings seen along the two roadsides. The building names are displayed on an iPad, which are projected onto the car windshield for the driver to observe, in a sense of augmenting the scene seen through the windshield by the projected building names. To implement such a system function, a PTZ camera and a two-camera omni-imaging device equipped on the vehicle roof are used for acquiring PTZ-images and omni-images, respectively, for the purposes of guidance map construction and vehicle localization. Guidance map construction is carried out by analyzing the PTZ-image to measure feature points on buildings. Vehicle localization during car driving is accomplished through 3D image analysis using omni-images acquired of a series of red circular-shaped landmarks attached on along-way objects. Good experimental results are shown to prove the feasibility of the proposed methods.

Keywords: *omni-image; augmented reality; tour guidance; guidance map; video surveillance vehicle*

1. INTRODUCTION

Nowadays, video cameras have become more and more popular in various applications, bringing great convenience in our daily life. In addition, the augmented reality technique can be of great help to tourists. Thirdly, in many cases touring by car in park areas is necessary. Therefore, it is desired in this study to integrate computer vision (CV) and augmented reality (AR) techniques to implement an augmented reality system for tour guidance using a video surveillance vehicle in outdoor park areas.

In recent years, many techniques have been developed for video surveillance applications. Trivedi et al. [1] proposed methods to enhance vehicle safety using omni-cameras and Kim and Suga [2] proposed a method to detect motion vectors using optical flows with an omni-camera. Moreover, new omni-vision systems can be designed by combining projective cameras and mirrors, and two omni-images together taken by such systems can provide stereo information. In this aspect, a method to obtain stereo information for mobile robot navigation with an omni-vision system consisting of two mirrors and one camera was proposed by He et al. [3]; and a method to detect suspicious passing persons by a vision system consisting of a pair of two-camera omni-imaging devices was proposed by Yuan and Tsai [4]. Jeng and Tsai [5] proposed a method based on the concept of space mapping to "calibrate" omni-cameras without computing their extrinsic camera parameters.

In addition, many methods for vehicle navigation using landmarks have been developed in the past. Betke and Gurvits [6] proposed a localization method to identify nearby landmarks and find their corresponding locations in a map. Wu and Tsai [7] proposed a vision-based method for location estimation for autonomous vehicle navigation in indoor environment using circular landmarks attached on ceilings. To detect landmarks and humans in omni-images, Ho and Chen [8] proposed an algorithm to detect ellipses, and Wang and Tsai [9] proposed a method to detect human faces by color and shape features using an elliptic skin color model.

In this study, an integration of augmented reality techniques and a video surveillance car system for park-area tour guidance is developed. In a similar application, Grosch [10] used panoramic images for navigation in real environments. Also, augmented reality for outdoor applications has been widely investigated in recent years. In this aspect, Lee et al. [11] conducted a study on using omni-vision to track, in large areas, the camera pose which simulates the user's view in an augmented reality environment; and Reitmayr and Drummond [12] proposed a model-based hybrid tracking system for outdoor AR uses enabling accurate and realtime overlays for a handheld device. In addition, AR techniques can assist car driving. For example, one can use the AR technique to create a head-up display (HUD) device in a car. Sandor et al. et al. [13] has proposed a method for doing this type of task.

An AR-based tour guidance system is developed in this study for use on a video surveillance vehicle as shown in Fig. 1(a). The main idea is that the names of the currently-visited buildings will be displayed in an iPad and projected on the car windshield as augmenting information to generate an AR effect as shown Fig. 1(b). The system has the following capabilities.

1. Laying circular-shaped red-colored landmarks on objects on sidewalks, and detecting automatically their locations using omni-images to estimate the relative position of the vehicle.
2. Creating local maps for respective landmarks, using omni-images acquired with a two-camera omni-imaging device as well as perspective-view images acquired by a PTZ camera, and converting the local maps into a global guidance map.
3. Determining the timing of passing near-by buildings, displaying their names on the iPad at right timings, and projecting the iPad screen onto the windshield to simulate an HUD effect for viewing by the driver to fulfill the tour guidance purpose.
4. Analyzing acquired omni-images to decide the vehicle turning direction so as to traverse the guidance map correctly according to guidance paths.

In the remainder of this paper, the ideas of the proposed methods and system is described in Section 2, the proposed methods for creating the guidance map, detecting circular-shaped landmarks for vehicle localization, conducting augmented reality based park guidance, and navigation by guidance map traversals are described in Sections 5 through 8, respectively, followed by some experimental results and conclusions in Sections 7 and 8, respectively.

2. IDEAS OF PROPOSED SYSTEM

2.1 System Architecture

The proposed system architecture for AR-based park-area guidance using a video surveillance vehicle is illustrated in Fig. 1(c). The two types of cameras, a two-camera omni-imaging device and a PTZ camera, used by the system and affixed on the top of the vehicle are shown in the figure. Also shown is a local network used by the system, in which an access point is constructed within the vehicle, and all the devices shown in the figure are connected to one another through it. A laptop computer COM_A is used for creating the guidance map and guiding the navigation process. It is also used to acquire not only two omni-images in each navigation cycle but also a PTZ-image using the PTZ camera. Another laptop computer COM_B is used to receive images from COM_A through the local network. An iPad is used to display the names of the currently-visited buildings for projection on the car windshield after receiving the information of the buildings from COM_A.

2.2 Learning Process

The proposed system operates in two phases — *learning* and *navigation*. In the learning process, the first work is to "calibrate" the cameras. For the two-camera omni-imaging device, we use the space-mapping method of Jeng and Tsai [5], and for the PTZ camera, we use a nonlinear angular mapping method proposed by Wang and Tsai [9]. More details will be described later.

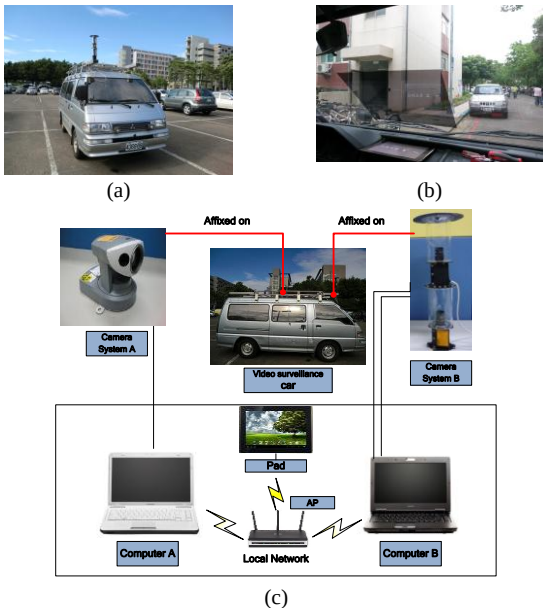


Fig. 1. Structure of proposed system. (a) Used video surveillance vehicle. (b) Building names displayed on iPad and projected onto windshield for viewing. (c) Architecture of proposed system.

The second work in learning is to create a *guidance map* for use in the navigation process. At first, the computer COM_A is used to analyze acquired omni-images and compute the location of each landmark by elliptical-shape fitting. Next, the computer COM_B is used to capture the PTZ-image of neighboring buildings, and feature points on them are selected manually and used to compute their distances with respect to the vehicle using the space-mapping method [5]. Also, through the aforementioned angular mapping method used in calibrating the PTZ-camera, we can get the orientations of the feature points as well. Afterwards, the distance and orientation data of the feature points are sent to COM_A to create a local map for the landmark. This process is repeated for all the landmarks. Finally, the generated local maps are integrated into a global one as a guidance map. More details will be described later.

2.3 Navigation Process

In each cycle of the navigation process by car driving, at first an omni-image is acquired and the existence of a landmark in the image is decided. If decided to be so, the space-mapping method is used again to compute the location of the landmark. Next, the turning direction of the vehicle is checked by an optical flow method [14] and used to track the vehicle in a graph representing the guidance map. Subsequently, the computed landmark location and the vehicle turning direction together are used to estimate the location of the vehicle. Accordingly, the nearby buildings are retrieved from the guidance

map. Finally, the building names are displayed on the iPad screen, which is then projected onto the car windshield to generate an AR effect for viewing by the driver. In this way, the driver can be well guided. More details will be introduced later.

3. CREATION OF GUIDANCE MAP

3.1 Calibration of PTZ-camera

The coordinate systems involved in the proposed system are shown in Fig. 2. The image coordinate system (ICS) defined for PTZ-images is described by image coordinates (u, v), and the spherical coordinate system (SCS) defined for the real-world space is described by the space coordinates (ρ, θ, φ). The SCS is a 3D polar coordinate system which can also be described by Cartesian coordinates (i, j, k), as shown in Fig. 2. That is, Cartesian coordinates (i, j, k) of a space point P may be transformed into spherical coordinates (ρ, θ, φ), and vice versa, where ρ is the distance between P and the origin S of the SCS; θ is the pan angle between the positive k-axis of the Cartesian coordinate system (CCS) and the line from S to P projected onto the ik-plane of the CCS; and φ is the tilt angle between the ik-plane and the line from S to P.

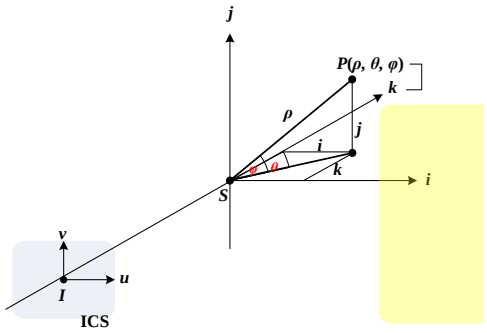


Fig. 2. Illustration of transformations between image coordinate system (ICS), spherical coordinate system (SCS), and Cartesian coordinate system (CCS).

In order to "calibrate" the PTZ camera for the purpose of computing the pan and tilt angles of space points which appear in the acquired PTZ-image, a grid board as shown in Fig. 3(a), which has m vertical lines and n horizontal lines, is attached on a wall perpendicular to the ground. Then, the angles between the PTZ-camera direction and the lines from the camera to the intersection points in the grid board are measured so that the pan and tilt angles of all the intersection points are known in advance. Then, an interpolation method is used to compute the pan and tilt angle values of each non-intersection point. These data finally are saved into a mapping table T_{PTZ} for later uses.

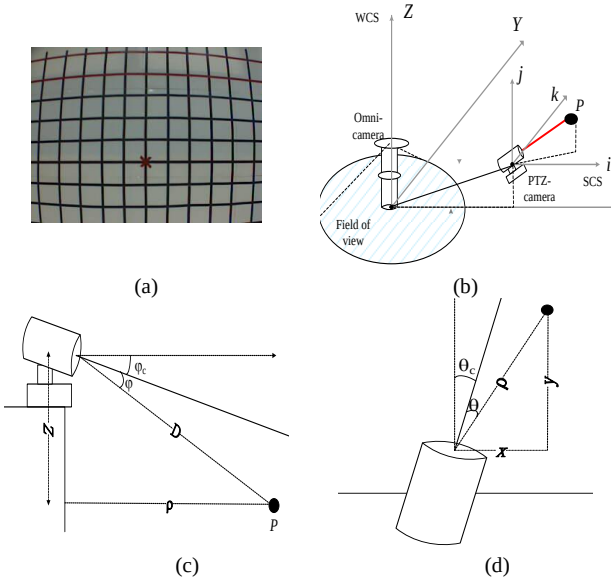


Fig. 3. PTZ-camera calibration and coordinate conversion. (a) Camera calibration by a grid board. (b) WCS which includes SCS. (c) A lateral view of a feature point in WCS. (d) A top view.

3.2 Conversion of PTZ-camera Coordinates of Feature Points into Omni-camera Coordinates

The omni-camera coordinate system (OCCS) is regarded as the world coordinate system (WCS) with coordinates (x, y, z), and the PTZ-camera coordinate system (PCCS) as the SCS, in this study. However, we also have the PTZ-camera in the WCS as shown in Fig. 3(b); so, in order to use the information of the PTZ-image coordinates of each pre-selected feature point P (like a corner point of a building), we have to transform these coordinates into the OCCS for use in creating the guidance map. For this aim, at first we acquire the tilt and pan angles φ_c and θ_c of the image point p in the PTZ-image corresponding to the pre-selected feature point P from the table T_{PTZ} created by the PTZ-camera calibration method described in Section 3.1 above.

Next, as illustrated in Fig. 3(c), we obtain the distance D between the PTZ-camera and point P by manual measurement. Then, the horizontal distance ρ between the PTZ-camera and P may be computed by:

$$\rho = D \times \cos(\varphi_c + \varphi)\,;\ z = D \times \sin(\varphi_c + \varphi)\,,$$

where φ, as shown in Fig. 3(c), is the lateral-view incident angle of P with respect to the optical axis of the PTZ-camera. Finally, the world coordinates (x, y, z) of P are computed as:

$$\rho = D \times \cos(\varphi_c + \varphi)\,;\ z = D \times \sin(\varphi_c + \varphi)\,;$$
$$x = \rho \times \sin(\theta_c + \theta) + L\,;\ y = \rho \times \cos(\theta_c + \theta) + K\,,$$

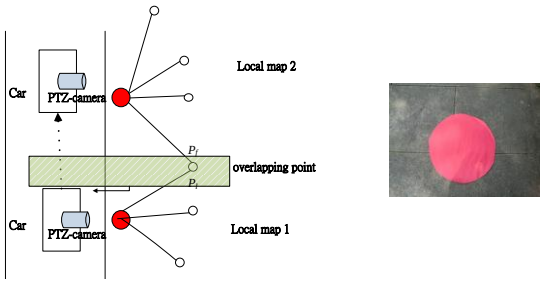
where θ, as shown in Fig. 3(d), is top-view incident angle of P with respect to the axis of the PTZ-camera; the parameters L and K are the respective positions of the PTZ-camera on the x-axis and y-axis of the WCS, both known in advance also by manual measurement.

3.3 Creation of Guidance Map

The guidance map displayed dynamically for viewing in each navigation cycle includes three types of data: the location of each landmark, the location of each building feature point, and the current location of the video surveillance car. In order to create the guidance map, we have to generate the local map of each

landmark at first, which includes data of the first two types for a landmark. To compute the location of each landmark P_{lm}, we take two omni-images I_1 and I_2 of P_{lm} by the two-camera omni-imaging devices and a perspective-view image I of P_{lm} by the PTZ-camera. Then, we detect the image points of landmark P_{lm} appearing in I_1 and I_2 and compute accordingly the coordinates of the center point of P_{lm} in the WCS according to a method proposed in the next section (Section 4). Next, we compute the location of each pre-selected feature point P_f on the buildings in the SCS according to the technique described previously in Section 3.2. Finally, we compute the location of P_f on the local map with respect to the landmark P_{lm}.

With the local maps of all the landmarks so created, we then integrate them to create a global guidance map. In this process, it is noted first that when we create the local maps, the last feature point of each local map and the first feature point of the next local map, as shown in Fig. 4(a), are taken to be an identical feature point. Therefore, each of such overlapping points can be connected to merge two neighboring local maps. Afterwards, we translate the locations of all the landmarks with respect to a *map coordinate system* (MCS) set up for the guidance map. Finally, we create the building borders by connecting every two neighboring feature points to create the guidance map.



(a) (b)

Fig. 4 Creation of global map and used landmark shape. (a) Merge of two local maps using an overlapping point. (b) Used landmark shape.

4. DETECTION OF CIRCULAR-SHAPED LANDMARK

4.1 Analysis of Landmark Shapes in Omni-images

In this study, we place red circular-shaped landmarks, as shown in Fig. 4(b), along tour paths for vehicle localization. The circular shape of each circular landmark in the omni-image may be approximated by an ellipse, as proved by Wu and Tsai [14]. Accordingly, as shown in Fig. 5, the hyperbolical shape of the mirror in the omni-camera may be described as:

$$\frac{R^2}{a^2}-\frac{z^2}{b^2}=-1,\quad R=\sqrt{x^2+y^2},\quad c=\sqrt{a^2+b^2},$$

where (x, y, z), and (u, v) respectively are the coordinates of the omni-camera coordinate system (OCCS) and the omni-image coordinate system (OICS); and the projection relationship between the omni-image coordinates (u, v) and the omni-camera coordinates (x, y, z) are described by [15]:

$$u=\frac{xf(b^2-c^2)}{(b^2+c^2)(z-c)-2bc\sqrt{(z-c)^2+x^2+y^2}},$$

$$v=\frac{yf(b^2-c^2)}{(b^2+c^2)(z-c)-2bc\sqrt{(z-c)^2+x^2+y^2}},$$

where the parameter f is the focal length of the omni-camera. The original circular shape in the OCCS of a landmark P_{lm} with a radius of R_w and its center located at (x_w, y_w, z_w) bay be described by $(x - x_w)^2 + (y - y_w)^2 = R_w^2$; $z = z_w$. Then, after rotating the two coordinate systems for the angle of $\theta_w = \tan^{-1}(y_w/x_w)$ as illustrated in Fig. 6, it can be proved that the following equation may be used to describe the approximate elliptical shape of a landmark in the omni-image:

$$\frac{(u'-u_w')^2}{R_w^2F'(x_w')^2}+\frac{v'^2M^2}{R_w^2}=1 \qquad (1)$$

where (u', v') are the new omni-image coordinates resulting from the rotation, (x_w', y_w', z_w') are the new omni-camera coordinates, and M and $F(x)$ respectively are:

$$M=\frac{f(b^2-c^2)}{(b^2+c^2)(z-c)-2bc\sqrt{(z-c)^2+x_w'^2}},$$

$$F(x)=\frac{xf(b^2-c^2)}{(b^2+c^2)(z-c)-2bc\sqrt{(z-c)^2+x^2}},$$

with $F'(x)$ being the first derivative of $F(x)$ with respect to x. Eq. (1) above describes an elliptical shape in the omni-image centered at $(u_w', 0)$ with the lengths of its major and minor axes being $R_wF'(x_w')$ and R_w/M, respectively.

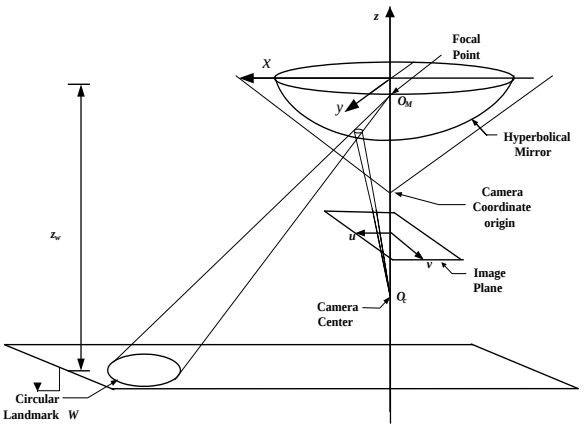


Fig. 5. Omni-camera coordinate system involved in this study.

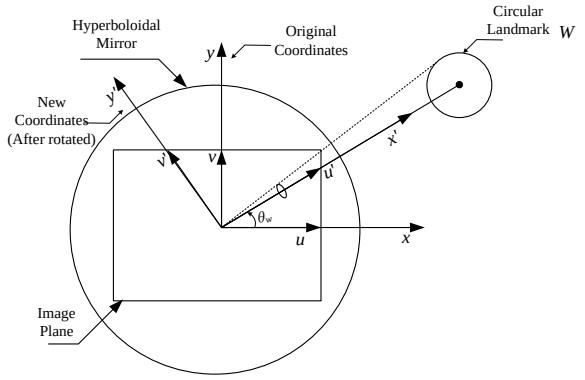


Fig. 6. Illustration of top view of a landmark.

In addition, the area of the elliptical shape can be shown to be as follows:

$$area = R_w{}^2 F'\left(x_w'\right)^2 \times \frac{R_w}{M} \times \pi \cdot \tag{2}$$

Because we use landmarks of the same size, the value R_w is a constant. In addition, the values $F'(x_w')$ and M are computed in terms of x_w', so if the areas of the ellipses are the same in omni-images, the distances x_w' will also be the same.

Finally, the distance u_w' between the center point of the ellipse and the image center in the omni-image may be shown to be:

$$u_w' = \frac{x_w' f(b^2 - c^2)}{(b^2 + c^2)(z - c) - 2bc\sqrt{(z - c)^2 + x_w'^2}}$$

from which we can see that if the areas of two ellipses are the same in the omni-image, the distance u_w' is also the same because x_w' has just been shown to be the same. Therefore, we can record the relationship between the area of the ellipse and the distance x_w' to simplify our landmark detection work. Furthermore, we may use least squares error fitting to express the relation between the area a of the elliptical shape and the distance u_w' with a curve described as follows:

$$u_w' = 3.080\,x^3 + 57690.503x^2 - 3879739x + 91113576 \tag{3}$$

where $x = 1/a$ and the coefficients (3.080, etc.) are obtained from fitting a set of pre-computed data of x and u_w' in landmark images.

Accordingly, when an ellipse is detected in an omni-image, we can obtain the area a of the ellipse and the value u_w'. Then, the value $x = 1/a$ may be substituted into the curve function

$$f(x) = 3.080\,x^3 + 57690.503x^2 - 3879739x + 91113576$$

described by the right-hand side of (3) above, and if the value $f(x)$ is checked to be larger than the value u_{wi}', then the ellipse in the omni-image may be regarded as a correctly detected circular-shaped landmark.

4.2 Red Circular-shaped Landmark Detection

To detect a red circular-shaped landmark in an omni-image, we use YCbCr colors, and let the Y value to range from 126 to 235; the Cr value to range from 220 to 240, and the Cb value be not limited in order to detect areas of red color. In the result of this color analysis process, noise might exist, which we remove by region growing and small area removal. Then, we use apply ellipse fitting to obtain the real elliptical shape and the center point of the ellipse in the omni-image. Finally, we conduct the circular-shaped landmark detection scheme described in the last section (Section 4.1) to see if a landmark really exists in the image.

4.3 Distance Estimation between a Surveillance Vehicle and a Circular-shaped Landmark

As described previously, we have to compute the location of each landmark (described by the coordinates of the landmark center in the WCS) with respect to the video surveillance vehicle during the process of guidance map generation. The detail of this process is described in this section.

A. Review of Adopted Space-mapping Method for 3D Data Acquisition

As illustrated in Fig. 7(a), to obtain the world coordinates (x, y, z), of a space point P from an omni-image taken by the two-camera omni-imaging device, finding two elevation angles α_1 and α_2 by looking up the pano-mapping tables of the two omni-cameras generated in the previously-mentioned space-mapping method is required [5]. In more detail, as shown in Fig. 7(b) the distance d between point P and the upper mirror center C_1 may be computed according to the triangulation principle as follows:

$$\frac{d}{\sin(90^\circ + \alpha_2)} = \frac{b}{\sin(\alpha_1 - \alpha_2)},$$

where the parameter b is the *baseline* of the omni-imaging device. The equation above may be reduced by trigonometry to be:

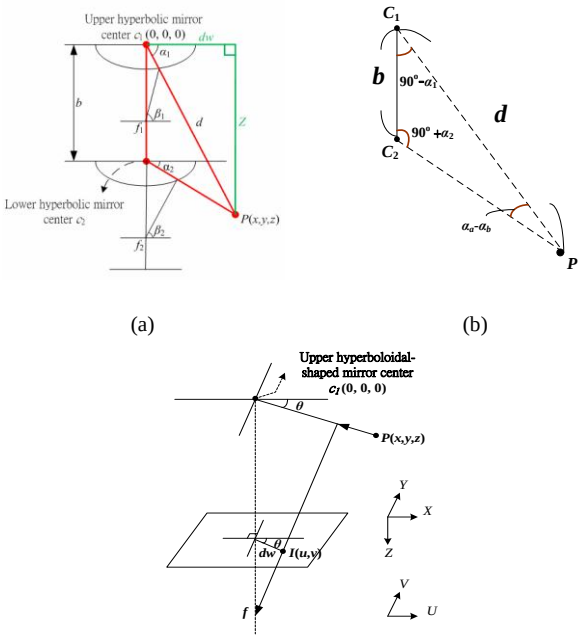
$$d = \frac{1}{\cos\alpha_1} \times \frac{b}{\tan\alpha_1 - \tan\alpha_2}.$$

As a result, the horizontal distance dw and the vertical distance z of P with respect to the omni-imaging device as shown in Fig. 7(a) may be computed by:

$$\begin{aligned} dw &= d\cos\alpha_1 = \frac{b}{\tan\alpha_1 - \tan\alpha_2}, \\ z &= d\sin\alpha_1 = \frac{b\tan\alpha_1}{\tan\alpha_1 - \tan\alpha_2}. \end{aligned} \tag{4}$$

Furthermore, assume that point P with world coordinates (x, y, z) is projected onto a point I with image coordinates (u, v) in the omni-image coordinate system (OICS). Then, according to the rotational invariance property of the omni-camera, the azimuth angle θ of P is just the angle of point I with respect to the u-axis in the OICS, as illustrated in Fig. 7(c). As a result, the azimuth angle can be computed by:

$$\theta = \cos^{-1}\frac{\mu}{\sqrt{u^2 + v^2}} = \sin^{-1}\frac{v}{\sqrt{u^2 + v^2}}. \tag{5}$$



(a) (b) (c)

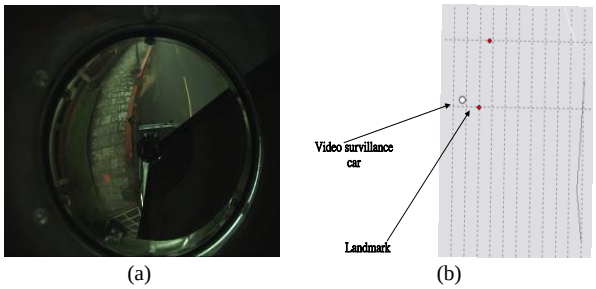
Fig. 7. Computation of depth. (a) Ray tracing of a scene point P imaged by omni-imaging device with a hyperboloidal-shaped mirror. (b) A triangle in detail, which is part of (a). (c) System configuration of upper omni-camera with a hyperbolical-shaped mirror.

Using (4) and (5), we can calculate the coordinates x and y by the distance dw and the azimuth angle θ in the WCS as follows:

$$x = dw \times \cos\theta = \frac{b\cos\theta}{\tan\alpha_1 - \tan\alpha_2}, \qquad (6)$$
$$y = dw \times \sin\theta = \frac{b\sin\theta}{\tan\alpha_1 - \tan\alpha_2}.$$

B. Calculation of Landmark Distance

After the 3D data (x, y, z) of a detected landmark represented by a space point P is obtained as described above, because the omni-camera coordinate system is regarded as the WCS and the guidance map is 2D in the WCS, we use such 3D data to compute further the distance D between the video surveillance vehicle and the landmark as $D = \sqrt{x^2 + y^2}$. We can then display part of the guidance map from the top view, as shown by the example of Fig. 8.



(a) (b)

Fig. 8. Relative position of a landmark and video surveillance car. (a) Omni-image. (b) Position of car on the guidance map.

5. PARK GUIDANCE BY AR TECHNIQUE

5.1 AR-based Guidance in a Vehicle with an iPad

In order to implement AR-based tour guidance in a park area via a video surveillance vehicle, we use an iPad to simulate an HUD device by projecting the image displayed on the iPad (called the iPad image hereafter) onto the car windshield. For this, we have to compute the correct position on the windshield in the WCS where the iPad image should be projected, as described next.

First of all, we introduce a *vehicle coordinate system* (VCS) for use on the video surveillance vehicle described by coordinates (x, y, z), as shown in Fig. 9. The origin O of the VCS is the lower right corner of the windshield and the xy-plane of the VCS is parallel to the ground. Let θ denote the angle between the windshield-plane and the xy-plane. Moreover, we place the iPad on the xy-plane. Then, the iPad image is projected onto the windshield to yield a *mirror image* as seen by the eyes of the driver. The angle between the mirror image plane and the windshield plane is also θ, as shown in Fig. 9(a). Let (x_e, y_e, z_e) denote the pre-measured coordinates of an eye p_e of the driver (assumed to be fixed during driving), and let (x_p, y_p, z_p) denote the coordinates of a point p_p in the iPad image, called an *iPad image point*, which is on the xy-plane. Let p_r denote the point in the mirror image corresponding to p_p, and let (x_r, y_r, z_r) denote its coordinates in the VCS. This *mirror point* p_r is formed by a light ray which comes from the iPad image point p_p, hits then onto a *projection point* p_w on the windshield, and is reflected to go finally into an eye represented by the *eye point* p_e. By the way, we consider the set of all projection points to compose a *projection image*, though it is non-existing physically. Now, we want to compute the coordinates (x_w, y_w, z_w) of the projection point p_w in the projection image on the windshield in the VCS for use later in determining the timing for showing the names of the buildings on the roadsides during driving. The computation process is described in the following.

At first, as shown in Fig. 9(a) it is not difficult to figure out that the included angle between the mirror image plane and the x-y plane is $180^o - 2\theta$, and that the normal vector of the windshield plane in the VCS is

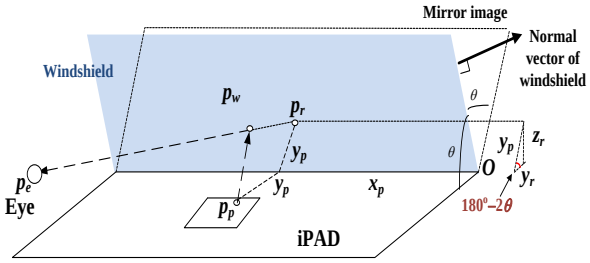
$$\vec{n} = [0, \sin(90^o - \theta), \cos(90^o - \theta)]^T. \qquad (7)$$

Accordingly, we can compute the coordinates (x_r, y_r, z_r) of mirror point p_r by

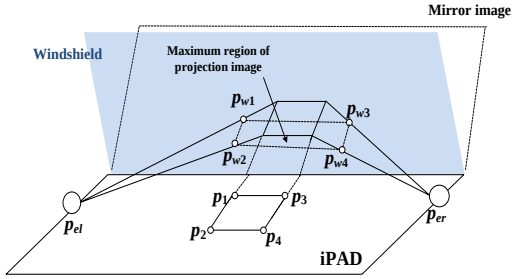
$$x_r = x_p,\ y_r = y_p\cos(180^o - 2\theta),\ z_r = y_p\sin(180^o - 2\theta). \qquad (8)$$

And regarding each point on the windshield plane P to be a vector $\vec{v} = [x, y, z]^T$ from the origin O, we have $\vec{n} \cdot \vec{v} = 0$ where "$\cdot$" is the inner product operator, because $\vec{n}$ and $\vec{v}$ are perpendicular. This leads to the following equation describing the windshield plane P:

$$y\times\sin(90^o - \theta) + z\times\cos(90^o - \theta) = 0. \qquad (9)$$



(a)



(b)

Fig. 9. Illustration of projection of iPad image onto car windshield. (a) Computing location of projection point p_w where p_r is mirror point corresponding to iPad image point p_p. (b) Illustration of maximum coverage of projection image enclosed by p_{w1} through p_{w4}.

Furthermore, we may describe the line L from eye point p_e to mirror point p_r by the following equation:

$$x = x_e + t(x_e - x_r),\ y = y_e + t(y_e - y_r),\ z = z_e + t(z_e - z_r) \quad (10)$$

where t is a parameter. The intersection of the plane P described by (9) and the line L described by (10) is just the projection point p_w whose coordinates may be obtained from solving the simultaneously equations of (9) and (10). For this, by substituting (10) into (9), we get a solution for t as follows:

$$t = -\frac{y_e \times \sin(90^o - \theta) + z_e \times \cos(90^o - \theta)}{(y_e - y_r) \times \sin(90^o - \theta) + (z_e - z_r) \times \cos(90^o - \theta)}.$$

And substituting the above solution for t into (10), we get finally the coordinates (x_w, y_w, z_w) of point p_w on as

$$x_w = x_e + t(x_e - x_r),\ y_w = y_e + t(y_e - y_r),\ z_w = z_e + t(z_e - z_r). \quad (10)$$

As illustrated by Fig. 9(b), using the above scheme we can compute, in the VCS, the locations of the projection points p_{w1} and p_{w2} corresponding to the left two corner points p_1 and p_2, respectively, of the iPad as viewed by the driver using an eye located at point p_{el}. Similarly, we can compute, in the VCS, the locations of the projection points p_{w3} and p_{w4} corresponding to the right two corner points p_3 and p_4, respectively, of the iPad as viewed by a guest sitting on the right-hand front seat of the vehicle using an eye located at point p_{er}. The four projection points p_{w1} through p_{w4} together form the maximum region of the projection image.

5.2 Showing Names of Buildings on the Windshield

In order to check the next landmark in the omni-image, we select a detection region in the omni-image, and classify it into the left-front region and the left-back region. Moreover, we can use the detected landmark and the car turning direction to estimate the position of the video surveillance vehicle on the guidance map. Accordingly, all the feature points of the nearby buildings within a certain distance can be retrieved from the guidance map. These points are described with coordinates in the WCS.

In order to display the names of the currently-visited buildings on the left and right road sides for AR-based guidance, at first in the VCS we compute the coordinates (x_{w1}, y_{w1}, z_{w1}) and (x_{w3}, y_{w3}, z_{w3}) of the leftmost and rightmost upper corner points p_{w1} and p_{w3} of the iPad image as described in the last section (Section 5.1). Then, we transform these coordinates into the WCS whose origin is the center of an omni-camera, resulting in the new coordinates $(x_{w1}', y_{w1}', z_{w1}')$ and $(x_{w3}', y_{w3}', z_{w3}')$ of p_{w1} and p_{w3}, respectively. Then, we describe the line l from p_{w1} to p_{w3} parametrically by

$$x = x_{w1}' + t(x_{w1}' - x_{w3}');\ y = y_{w1}' + t(y_{w1}' - y_{w3}'), \quad (11)$$

where t is parameter supposed to be in the range [0, 1].

Next, as shown in Fig. 10 we create building borders using the retrieved nearby building feature points mentioned above, and for each building border l_j with its two end points at coordinates (x_j, y_j, z_j) and $(x_{j+1}, y_{j+1}, z_{j+1})$, respectively, in the WCS, we describe it parametrically as well by

$$x = x_j + s(x_j - x_{j+1});\ y = y_j + s(y_j - y_{j+1}), \quad (12)$$

where s is a parameter supposed to be in the range [0, 1], too. Finally, we solve (11) and (12) to get a solution for s as follows:

$$s = \frac{(y_{w3}' - y_{w1}')(x_j - x_{w1}') + (x_{w3}' - x_{w1}')(y_{w1}' - y_j)}{(x_{w3}' - x_{w1}')(y_{j+1} - y_j) - (y_{w3}' - y_{w1}')(x_{j+1} - x_j)}. \quad (13)$$

Then, if s is in [0, 1], then it means that the two lines l and l_j described by (10) and (11) indeed have an intersection, and it in turn means that the vehicle has just passed the building and the name of the building should be displayed right away in the iPad image for projection on the windshield.

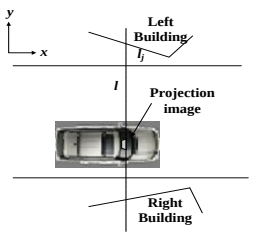
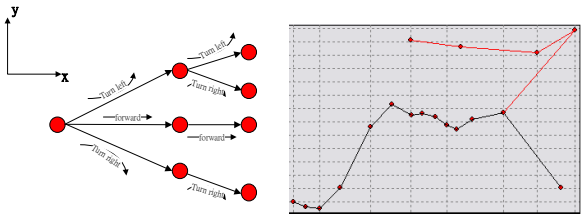


Fig. 10. Checking timing for display building names.

6. NAVIGATION BY MAP TRAVERSALS USING VEHICLE TURNING DIRECTIONS

When meeting a branching road in a guidance tour, it is desired to analyze the motions in the consecutively acquired omni-images to determine the vehicle turning direction. Chen and Tsai [14] applied optical flow analysis to implement this idea, which is adopted in this study. Moreover, in order to detect the direction of the vehicle movement, right, left, or forward, on the guidance map, we have to organize the guidance map into a graph which we call the *guidance map graph*. At first, we have to specify the relationship between the label of the current landmark and that of the previous landmark like the tree shown in Fig. 11(a). Using the labels of the landmarks of the three directions, we can connect the landmarks to construct a guidance map graph, like that shown in Fig. 11(b). Such a guidance map graph may be used to generate a *guidance path* for each tour. Accordingly, we can analyze the omni-image to decide the vehicle turning direction during navigation so that the system can "traverse" the guidance map graph correctly according to the guidance path.



(a) (b)

Fig. 11 Graph of guidance map. (a) Organizing guidance map. (b) Guidance map graph of experimental environment.

7. EXPERIMENTAL RESULTS

In this section, we show some experimental results of applying the proposed system for park touring in part of the campus of National Chiao Tung University which is illustrated in Fig. 12. At first, the result of creating the guidance map for the experimental environment of Fig. 12 is shown in Fig. 13. Then, three results of a navigation process based on a guidance path are shown in Figs. 14 through 16. Many more successful navigation sessions have been conducted in the experimental environment, showing the feasibility of the proposed system.

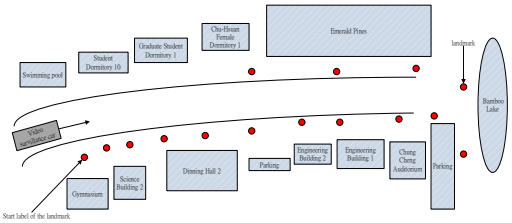


Fig. 12. Illustration of experimental guidance area.

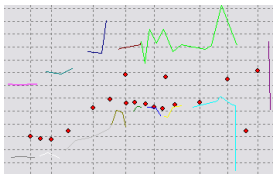
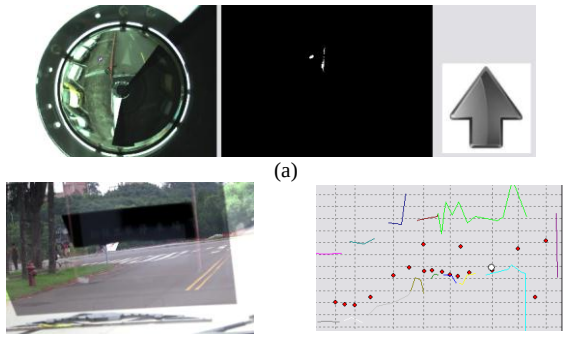


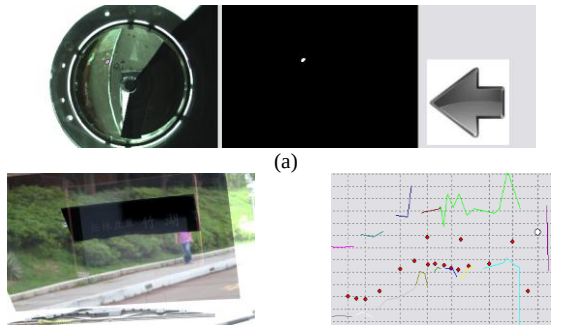
Fig. 13. Generated guidance map of experiment environment.



(a)

(b) (c)

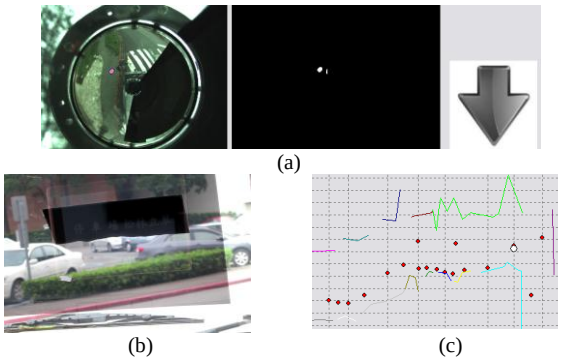
Fig. 14. An experimental result. (a) Omni-image, image of detection result, and decided vehicle direction. (b) iPad image projected on windshield. (c) Position of vehicle on guidance map.



(a)

(b) (c)

Fig. 15. A 2nd experimental result. (a) Omni-image, image of detection result, and decided vehicle direction. (b) iPad image projected on windshield. (c) Position of vehicle on guidance map.



(a)

(b) (c)

Fig. 16. A 3rd experimental result. (a) Omni-image, image of detection result, and decided vehicle direction. (b) iPad image projected on windshield. (c) Position of vehicle on guidance map.

8. CONCLUSIONS

An AR-based tour guidance system in park areas via a video surveillance vehicle and computer vision techniques has been proposed. While driving the vehicle in a park, a user can get from the system tour guidance about the names of the buildings appearing along the two roadsides. The building names are displayed on the

screen on an iPad which is then projected onto the windshield for the driver to observe. The effect of this projection is equivalent to augmenting the road scene seen by the driver with the projected building names. Experimental results show the feasibility of the proposed system. Future studies may be directed to implementing the AR effect on an iPad, instead on the windshield and using more automated methods for vehicle localization.

REFERENCES

[1] T. Gandhi and M. M. Trivedi, "Motion analysis for event detection and tracking with a mobile omni-directional camera," *ACM Multimedia Systems Journal, Special Issue on Video Surveillance*, vol. 10, no. 2, pp. 131–143, 2004.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. of 7th Int'l Joint Conf. on Artif. Intell.*, Vancouver, Canada, pp. 674 – 679, 1981.

[3] L. He, C. Luo, F. Zhu, Y. Hao, J. Ou and J. Zhou, "Depth map regeneration via improved graph cuts using a novel omnidirectional stereo sensor," *Proc. 11th IEEE Int'l Conf. on Computer Vision*, Rio de Janeiro, Brazil, pp. 1-8, 2007.

[4] P. H. Yuan, K. F. Yang, and W. H. Tsai, "A Study on Monitoring of Nearby Objects around a Video Surveillance Car with a Pair of Two-camera Omni-directional Imaging Devices," *Proc. of 2010 Int'l Computer Symp.*, Tainan, Taiwan, pp. 325-330, Dec. 2010.

[5] S. W. Jeng and W. H. Tsai, "Using pano-mapping tables for unwarping of omni-images into panoramic and perspective-view images," *J. of IET Image Processing*, Vol. 1, No. 2, pp. 149-155, June 2007

[6] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, Vol. 13, No.2, pp. 251-263, April 1997.

[7] C. J. Wu, "New Localization and Image Adjustment Techniques Using Omni-Cameras for Autonomous Vehicle Applications," *Ph. D. Dissertation*, Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, July 2009.

[8] C. T. Ho and L. H. Chen, "A high-speed algorithm for elliptical object detection," *IEEE Trans. on Image Processing*, Vol. 5, No. 3, pp. 547-550, March 1996.

[9] Y. T. Wang and W. H. Tsai, "Indoor security patrolling with intruding person detection and following capabilities by vision-based autonomous vehicle navigation," *Proc. of 2006 Int'l Computer Symp., Int'l Workshop on Image Processing, Computer Graphics, and Multimedia Technologies*, Taipei, Taiwan, Dec. 2006

[10] T. Grosch, "PanoAR: Interactive Augmentation of Omni—Directional Images with Consistent Lighting," *Proc. Of Mirage 2005, Computer Vision/Computer Graphics Collaboration Techniques and Applications*, Univ. of Koblenz-Landau, Germany, pp. 25-34, 2005.

[11] J. W. Lee, S. You and U. Neumann, "Tracking with Omni-Directional Vision for Outdoor AR Systems," *Proc. of IEEE ACM Int'l Symposium on Mixed & Augmented Reality*, Darrnstadtt, Germany, Oct. 2002.

[12] G. Reitmayr and T. W. Drummond, "Going out: Robust model based tracking for outdoor augmented reality," *Proc. IEEE Int'l Symp. Mixed and Augmented Reality (ISMAR)*, Santa Barbara, California, USA, pp. 109–118, 2006.

[13] M. Tonnis, C. Sandor, G. Klinker, C. Lange and H. Bubb. "Experimental evaluation of an augmented reality visualization for directing a car driver's attention," *Proc. of IEEE & ACM Int'l Symp. on Mixed & Augmented Reality*, pp. 56–59, Vienna, Austria, Oct. 2005.

[14] C. F. Chen and W. H. Tsai, "A Study on Surrounding Environment Monitoring by a Video Surveillance Car with Two 2-camera Omni-imaging Devices," *Proc. of 2012 Conf. on Computer Vision, Graphics & Image Processing*, Chiayi, Taiwan, 2012.

[15] H. Koyasu, J. Miura, and Y. Shirai, "Recognizing moving obstacles for robot navigation using real-time omni-directional stereo vision," *J. of Robotics and Mechatronnics*, Vol. 14, No. 2, pp. 147-156, June 2002.