Real-time Security Monitoring around a Video Surveillance Vehicle with a Pair of Two-camera Omni-imaging Devices

Pei-Hsuan Yuan, Kuo-Feng Yang and Wen-Hsiang Tsai, *Senior Member*, *IEEE*

***Abstract*—A pair of two-camera omni-imaging devices is designed for use on the roof of a video surveillance vehicle and corresponding 3D vision-based techniques for real-time security surveillance around the vehicle are proposed, which may be used for monitoring passing-by persons around the vehicle. First, the design of the pair of two-camera omni-imaging devices, each device consisting of two omni-cameras with their optical axes vertically aligned, is described. Then, a new analytic technique for fast 3D space data acquisition based on a pano-mapping method for image-to-world space transformation and the rotational invariance property of the omni-image is proposed. Also proposed are techniques for constructing top-view and perspective-view images for convenient observation of the monitored environment. Finally, 3D vision techniques for detecting automatically passing-by persons and computing their locations and body heights are proposed, followed by experimental results showing the precision and feasibility of the proposed techniques.**

***Index Terms*—video surveillance, omni-camera, omni-image, 3D data acquisition, passing-by person detection, security monitoring, top-view image, perspective-view image.**

I. Introduction

Video surveillance of environments has been studied intensively in recent years [1, 2]. In order to increase the mobility of the video surveillance system, vehicles are used as the carriers of such systems [3, 4, 5]. Applications of video surveillance vehicles include dynamic monitoring of outdoor events, detection of passing-by persons, assistance for safe driving, warning of dangerous activities, watching of environmental changes, etc. Various types of cameras were used to capture environment images. Gandhi and Trivedi [3] made a good survey of vehicle surround capture techniques and proposed a novel omni-video based approach for synthesizing dynamic panoramic surround maps using stereo and motion analysis of video images from a pair of omni-cameras on a vehicle. Micheloni, et al. [4] used an autonomous vehicle to monitor moving objects in indoor environments, Chen and Tsai [5] designed an autonomous vehicle to monitor planar objects on walls in buildings, and both works used projective cameras to capture environment images. Onoe et al. [6] and Mituyosi et al. [7] used omni-cameras for tracking human body features. A video surveillance system for localizing objects using multiple omni-cameras was proposed by Morita et al. [8]. Some related works using pairs of omni-cameras with hyperboloidal-shaped reflective mirrors can be found in [9, 10]. Specifically, Koyasu et al. [9] proposed an omni-directional stereo system consisting of two vertically aligned omni-cameras to detect and track obstacles. And Ukida et al. [10] used a similar system and a space encoding scheme to acquire 3D environment data for various applications. Furthermore, a method that reconstructs the 3D data of static nearby vehicles by a mobile robot using a stereo omni-camera (a two-mirror omni-imaging system) was proposed by Meguro et al. [11]. Many more related techniques can be found in [15-27], which will be reviewed after presenting the method proposed in this study.

Most of the above-mentioned works are about indoor visual surveillances and the speeds for computing the 3D data were generally slow. Some of the works used autonomous vehicles to carry the camera system. Very few studies about security monitoring were reported to use a *video surveillance vehicle* — a car equipped with a camera system on its roof. In this study, it is desired to design a video surveillance vehicle of this type equipped with an omni-imaging system for real-time wide-area monitoring applications, which has the following capabilities: (1) constructing a top-view image of the video surveillance vehicle's surrounding area for convenient observation of the environment security; (2) constructing a perspective-view image of any direction automatically or at any time instant specified by the user for real-time inspection; (3) detecting and displaying any suspicious passing-by person automatically; and (4) measuring the 3D features (location and height) of the passing-by person for security investigation purposes.

More specifically, for the specific purpose of security monitoring around a video surveillance vehicle, we design in this study an *omni-imaging device* which is composed of two

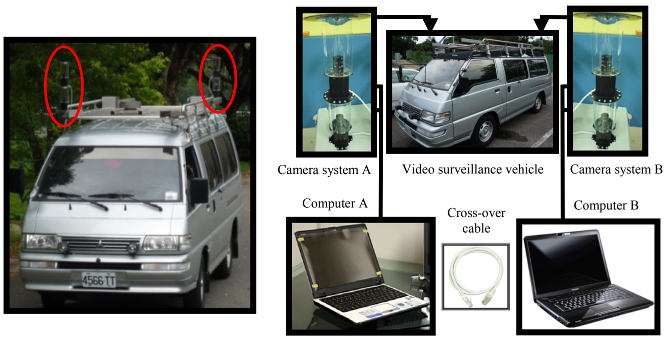
Manuscript received October 16, 2010. This work was supported by the NSC project No. 98-2221-E-009-116-MY3.

Pei-Hsuan Yuan was with the Institute of Multimedia Engineering, National Chiao Tung University, Hsinchu, Taiwan 300 and is now with Taiwan Semiconductor Manufacturing Company , Ltd., Hsinchu, Taiwan, 300 (e-mail: yuanegg.cs97g@nctu.edu.tw).

Kuo-Feng Yang is with the Institute of Computer Science and Engineering, and Wen-Hsiang Tsai is with the Dept. of Computer Science, both at National Chiao Tung University, Hsinchu, Taiwan 300 (e-mails: castle7322.cs97g @nctu.edu.tw, whtsai@cis.nctu.edu.tw).

catadioptric cameras with their optical axes vertically aligned, and affix a pair of such omni-imaging devices to the roof of a vehicle to acquire the environment data. One device is affixed to the right-front corner of the vehicle's roof and the other to the left-rear corner (see Fig. 1(a)). A new method, which combines skillfully the uses of such a pair of omni-imaging devices, a so-called pano-mapping technique [12], as well as the rotational invariance property of the omni-image to compute the 3D data of real-world points, is then proposed. The computation is based on *table lookup* and *analytic formulas* and so can be carried out in real-time to construct both top- and perspective-view images for instantaneous observation of the vehicle's surrounding environment. Also, a vision-based scheme using a vertical-line property in the omni-image for detecting passing-by persons, computing their locations and heights, and displaying such feature data both from the top and the perspective views for convenient inspection is proposed as well, which are not found in previous works. A detailed comparison of the proposed method with related existing methods will be given later.

In the remainder of this paper, the configuration of the proposed system, the design of the omni-camera system, and the proposed technique for computing 3D data are described in Section II. The proposed techniques for detection and monitoring of a passing-by person are described in Section III. And the schemes proposed for generating top- and perspective-view images are described in Section IV, followed by some experimental results and a comparison of the proposed method with related methods in Section V. Finally, conclusions and descriptions of the contributions made in this study are given in Section VI.



(a) (b)

Figure 1. Video surveillance vehicle and network used in this study. (a) Vehicle with omni-cameras (circled in red). (b) Network connecting computers.

II. System Configuration and Design of Proposed Omni-imaging Devices

A. System Configuration

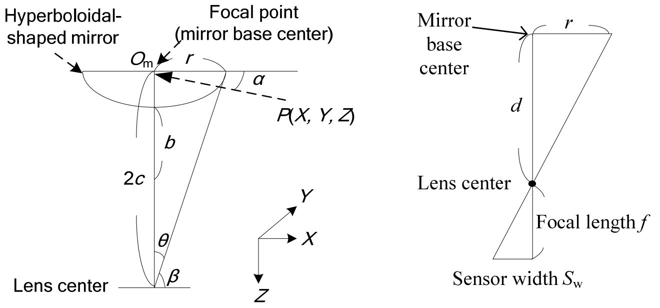
As illustrated in Fig. 1(b), the proposed system used in this study includes a video surveillance vehicle, a pair of two-camera omni-imaging devices CS_A and CS_B affixed on the vehicle's roof, and two laptop computers COM_A and COM_B inside the vehicle. Each two-camera omni-imaging device is controlled by a laptop computer, and a local network was designed to connect the two computers. Specifically, COM_A is used to display the top-view image of the surrounding area of the video surveillance vehicle, and COM_B to display the perspective-view images of specified directions outside the surveillance vehicle. The system process is divided into two phases, the *learning* phase and the *patrolling* phase. In the former, the pano-mapping tables of the used omni-cameras are constructed in advance, and in the latter, the video surveillance vehicle is driven in the outdoor environment for real security monitoring applications.

B. Camera Design Principle

The structure of each catadioptric omni-camera with a reflective mirror of the hyperboloidal shape used in this study is illustrated in Fig. 2, where the *world coordinate system* (WCS) is specified by (X, Y, Z) with its origin O_m located at the mirror base center which we assume to be coincident with one focal point of the hyperboloidal shape of the mirror. The shape of the mirror in the *camera coordinate system* (CCS), which is specified by (x, y, z) with its origin O_a located at the middle point between the point O_m and the lens center of the projective camera, may be described [12] as

$$\frac{s^2}{a^2}-\frac{z^2}{b^2}=-1, \qquad s=\sqrt{x^2+y^2} \tag{1}$$

where a and b are two parameters of the hyperboloidal shape. The parameter d, as shown in Fig. 2(b), is the distance between the camera lens center and the mirror base center, whose value can be obtained by a simple formula $d = 2c$ with $c^2 = a^2 + b^2$. Also, the axis of the camera is aligned with that of the mirror, and the lens center is fixed at the other focal point of the mirror shape.



(a) (b)

Figure 2. Omni-camera structure. (a) Omni-camera geometry. (b) Geometry between hyperboloidal-shaped mirror and projective camera.

By the shape geometry of a hyperboloid described by (1), the value α specifying the elevation angle of a real-world point P shown in Fig. 2(a) can be computed [10] by:

$$\tan\alpha=\frac{(b^2+c^2)\sin\beta-2bc}{(b^2-c^2)\cos\beta} \tag{2}$$

where the angle β as shown in Fig. 2(a) can be computed as:

$$\beta = \pi/2 - \theta \tag{3}$$

with

$$\theta = \tan^{-1}(r/2c) \tag{4}$$

where r is the distance of P to O_m on the XY plane and equals the radius of the circular mirror base when $\alpha = 0$. In Fig. 2(b), by the principle of similar triangles, we have

$$\frac{d}{r}=\frac{f}{S_w} \tag{5}$$

where f is the focal length of the projective camera and S_w is the width of the square CMOS sensor in the camera.

The goal of omni-camera design in this study is to construct a mirror of the hyperboloidal shape and determine the distance from the camera to the mirror under the constraint that f, S_w, and r are of given values which fits the structure of the vehicle roof. The projective camera we use has $f = 6$ mm and $S_w = 2.4$ mm, and to affix the cameras to a steel rod on the vehicle roof, we chose the mirror to have a base with an appropriate radius r of 4 cm. So, according to (5) and the previously-mentioned fact that $d = 2c$, we can derive d and c respectively to be $d = 10$ cm and $c = 5$ cm. Also, according to (3) and (4) and with α assumed to be zero, the values of the angles θ and β can be computed to be $\theta = 21.8^{o}$ and $\beta = 68.2^{o}$. Finally, using β and c, we may reduce (2) to be an equation with only one variable b of the following form

$$(b^2 + 25)\times 0.9285 - 10\times b = 0$$

which may then be solved to get $b = 3.39$ cm. And by $c^2 = 5^2 = a^2 + b^2$, a can be solved to be 3.68 cm. Thus, all of the desired parameters of the hyperboloidal-shaped mirror are obtained: $a = 3.68$, $b = 3.39$, $c = 5$, $d = 10$, all in the unit of cm.

C. 3D Data Acquisition

Each omni-imaging device used in this study consists of two omni-cameras designed in the way described above. The two cameras, one called the *upper* and the other the *lower*, are tied together, with their mirrors both facing down and their axes vertically aligned. An alternative illustration of the upper omni-camera configuration is shown in Fig. 3, where the *image coordinate system* (ICS) is specified by (u, v) with the image center as its origin, and a pixel p with image coordinates (u, v) corresponds to a real-world point P with coordinates (X, Y, Z).

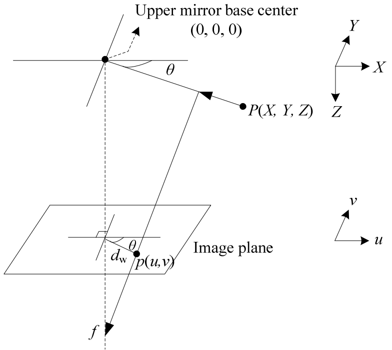


Figure 3. Configuration of an upper omni-camera with a mirror.

In this study, a new technique is proposed to compute the 3D data, (X, Y, Z), of each real-world point P in the WCS by the use of the two elevation angles α_1 and α_2 of P with respect to the mirror bases of the two omni-cameras in an omni-imaging device, as illustrated in Fig. 4, where the values of α_1 and α_2 may be obtained by using a pano-mapping method proposed by Jeng and Tsai [12].

More specifically, as shown in Figs. 3 and 4(a), each image pixel p is the projection of a real-world point P whose light ray goes onto the mirror of the omni-camera and is then reflected onto the camera imaging plane, resulting in an azimuth angle θ (shown in Fig. 3) and an elevation angle α (shown in Fig. 4(a) as α_1 or α_2). By the pano-mapping method proposed in [12], given the image coordinates (u, v) of an image pixel p corresponding to a real-world point P, the elevation angle α and the azimuth angle θ of P can be obtained by table lookup using a *pano-mapping table*. An example of the pano-mapping table is shown in Table 1. Also, according to the mirror surface geometry which we assume as usual to be radially symmetric, a relationship, called the *radial stretching function* and denoted as f_s, between the elevation angle α and the radial distance r of p in the image plane with respect to the image center is established to be:

$$r = f_s(\alpha) = a_0 + a_1\times\alpha^1 + a_2\times\alpha^2 + \ldots + a_5\times\alpha^5$$

where the coefficients a_0 through a_5 are computed in the following way using the *known* image coordinates (u_i, v_i) and the corresponding *known* world coordinates (X_i, Y_i, Z_i) of six real-world *landmark points* P_i selected manually in advance, where $i = 1$ through 6: (1) compute the radial distances r_i for each of the six points as $r_i = \sqrt{u_i^2 + v_i^2}$; (2) compute the elevation angle α_i for each of the six points as $\alpha_i = \tan^{-1}(Z_i/\sqrt{X_i^2 + Y_i^2})$; (3) solve the following six simultaneous equations to get the values of a_0 through a_5:

$$r_i = f_s(\alpha_i) = a_0 + a_1\times\alpha_i^1 + a_2\times\alpha_i^2 + \ldots + a_5\times\alpha_i^5$$

where $i = 1, 2, \ldots, 6$.

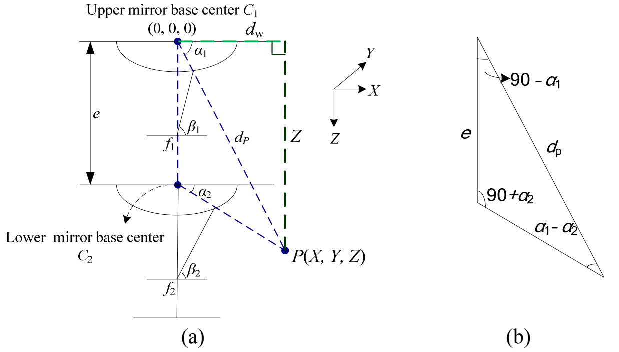


Figure 4. Computing 3D data. (a) Ray tracing of a real-world point P in an imaging device. (b) Blue triangle in (a) shown in detail.

Table 1. An example of pano-mapping table of size $M\times N$ [12].

	θ_1	θ_2	θ_3	θ_4	…	θ_M
α_1	(u_{11}, v_{11})	(u_{21}, v_{21})	(u_{31}, v_{31})	(u_{41}, v_{41})	…	(u_{M1}, v_{M1})
α_2	(u_{12}, v_{12})	(u_{22}, v_{22})	(u_{32}, v_{32})	(u_{42}, v_{42})	…	(u_{M2}, v_{M2})
α_3	(u_{13}, v_{13})	(u_{23}, v_{23})	(u_{33}, v_{33})	(u_{43}, v_{43})	…	(u_{M3}, v_{M3})
α_4	(u_{14}, v_{14})	(u_{24}, v_{24})	(u_{34}, v_{34})	(u_{44}, v_{44})	…	(u_{M4}, v_{M4})
…	…	…	…	…	…	…
α_N	(u_{1N}, v_{1N})	(u_{2N}, v_{2N})	(u_{3N}, v_{3N})	(u_{4N}, v_{4N})	…	(u_{MN}, v_{MN})

With a_0 through a_5 derived, a pano-mapping table like that of Table 1 is then constructed in the following way: for each real-world point P_{ij} with azimuth-elevation angle pair (θ_i, α_j), compute the coordinates (u_{ij}, v_{ij}) of the corresponding image pixel p_{ij} as

$$u_{ij} = r_j\times\cos\theta_i;\ \ v_{ij} = r_j\times\sin\theta_i$$

where $r_j = f_s(\alpha_j) = a_0 + a_1\times\alpha_j^1 + a_2\alpha_j^2 + \ldots + a_5\times\alpha_j^5$. After the table is constructed, when an image pixel p with coordinates (u, v) in a given omni-image is given and checked by table lookup to be located in entry E_{ij} with coordinates (u_{ij}, v_{ij}) in Table 1, the azimuth-elevation angle pair of the corresponding real-world point P can be obtained to be (θ_i. α_j). Note that this azimuth-elevation angle pair only says that P is located on a light ray R with the azimuth direction θ_i and the elevation angle α_j; it does not specify the 3D position of P sufficiently — any real-world point on light ray R will appear identically to be the pixel p in the image with the same coordinates (u_{ij}, v_{ij}). Also note that the pano-mapping table involves no camera parameter and is invariant in nature with respective to the camera position (i.e., it is not changed when the camera is moved around).

Back to the discussion about computing the 3D coordinates for a real-world point P in the WCS. Since two omni-cameras are used in each imaging device, after the two image pixels p_1 and p_2 corresponding to real-world point P are identified in the two omni-images taken by the two cameras, two elevation angles α_1 and α_2 as shown in Fig. 4(a) can be obtained by the above table lookup process using the image coordinates (u_1, v_1) and (u_2, v_2) of p_1 and p_2, respectively. With the upper mirror base center being located at the WCS origin with world coordinates (0, 0, 0), the goal of 3D data computation now can be achieved by using α_1 and α_2 to compute the world coordinates (X, Y, Z) of P. For this, as shown in Fig. 4(b), by the law of sines in trigonometry, we have:

$$\frac{d_P}{\sin(90+\alpha_2)} = \frac{e}{\sin(\alpha_1-\alpha_2)} \qquad (7)$$

where d_P is the distance between P and the base center of the upper mirror C_1, and e is the disparity between the two cameras in the omni-imaging device, which is measured manually in advance. Eq. (7) may be reduced to get d_P as:

$$d_P = \frac{1}{\cos\alpha_1}\times\frac{e}{\tan\alpha_1-\tan\alpha_2}. \qquad (8)$$

And so the horizontal distance d_w and the vertical distance Z of P as shown in Fig. 4(a) may be computed respectively to be:

$$d_\text{w} = d_P\cos\alpha_1 = \frac{e}{\tan\alpha_1-\tan\alpha_2};$$
$$Z = d_P\sin\alpha_1 = \frac{e\times\tan\alpha_1}{\tan\alpha_1-\tan\alpha_2}. \qquad (9)$$

Furthermore, according to Fig. 3, the azimuth angle θ in the figure can be described in terms of the pixel coordinates (u, v) as follows:

$$\sin\theta = \frac{v}{\sqrt{u^2+v^2}};\ \cos\theta = \frac{u}{\sqrt{u^2+v^2}} \qquad (10)$$

from which the value of θ in the ICS can be computed.

Now, according to the characteristic that the axes of the cameras are aligned with the axis of the mirror, the rotational invariance property of the omni-camera says that the azimuth angle of point P in the WCS and the azimuth angle of the corresponding pixel p in the ICS are identical [12]. Denoting both of the angles by θ, we can compute the values of X and Y in the WCS as:

$$X = d_\text{w}\times\cos\theta = \frac{e\times\cos\theta}{\tan\alpha_1-\tan\alpha_2}; \qquad (11)$$

$$Y = d_\text{w}\times\sin\theta = \frac{e\times\sin\theta}{\tan\alpha_1-\tan\alpha_2}. \qquad (12)$$

In summary, given a pair of matching image points corresponding to a real-world point P, we can compute the azimuth θ of P by Eqs. (10), and obtain also a pair of elevation angles α_1 and α_2 of P by pano-mapping table lookup. Then, the world coordinates (X, Y, Z) describing the unique 3D position of P can be computed by Eqs. (9), (11), and (12) *analytically*.

III. Automatic Detection of Passing-by Persons

A. Detection of Moving Objects in Omni-images

Before extracting moving objects around a surveillance vehicle, *background images* without objects are captured in advance in the learning phase with the vehicle in a static state, or, whenever necessary, in the patrolling phase with the vehicle also in a static state. An example is shown in Fig. 5. Then, *foreground images* possibly with objects are taken in real-time in the patrolling phase. Both background and foreground images are color ones, which are transformed into grayscale ones at the beginning of the object detection process. By subtracting the background image from the foreground one, a *difference image* is obtained. Because there usually exist noise pixels in the difference image, such as those caused by light variations, the difference image is thresholded next into a binary one to eliminate such noise pixels using the moment-preserving thresholding technique proposed by Tsai [13]. After these steps, we can obtain a bi-level image I_bl with detected moving objects all labeled as black pixels. An example of the result is shown in Fig. 6.

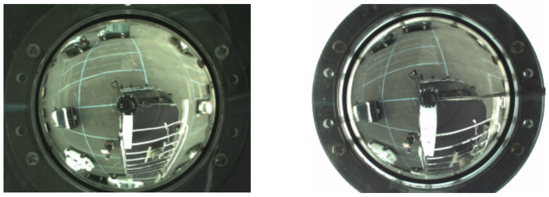

(a) (b)

Figure 5. Background images taken by a two-camera imaging device. (a) Background taken by upper camera. (b) Background taken by lower camera.

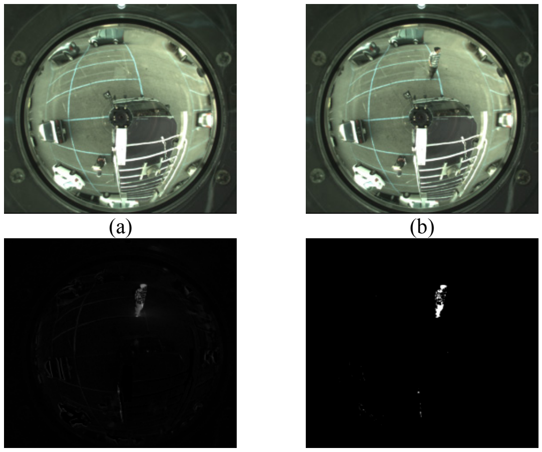

(a) (b)
(c) (d)

Figure 6. Passing-by person detection. (a) Background image. (b) Foreground image. (c) Difference image obtained by subtracting (b) from (a) (d) Binary image obtained by bi-level thresholding.

B. Detection of a Passing-by Person's Head

As illustrated in Fig. 7, the proposed technique to detect a passing-by person's head is based on a vertical-line property of the omni-image: each line L, which is parallel to the Z-axis in the WCS and so vertical to the ground, is projected onto the image as a line l_L *going through the image center*. This means that if a passing-by person stands on the ground, the *midline* l_L of the person's silhouette will go through the image center O_c, as illustrated in Fig. 8(a). So, we may find the top of each passing-by person's head by the following steps: (1) transform the rectangular image coordinates (u, v) of the bi-level difference image I_{bl} into polar ones (θ, r) where θ is the azimuth angle and r the radial distance; (2) scan each radial line inward from the image boundary to the image center based on the use of (θ, r); (3) find each *sufficiently-long* line segment in I_{bl} and group neighboring segments so found into a set L_{s}; (4) find the farthest line-segment end point in L_{s} with respect to the image center as the passing-by person's head location. If non-human objects will appear possibly in the omni-image, Step (3) should be refined using features like the *width* of the set L_{s} (equal to the number of line segments in L_{s} which means the width of the person) to differentiate and exclude them. The result of applying this process to Fig. 6 is shown in Fig. 8(b). Another result of detecting two persons is shown in Figs. 8(c) and 8(d).

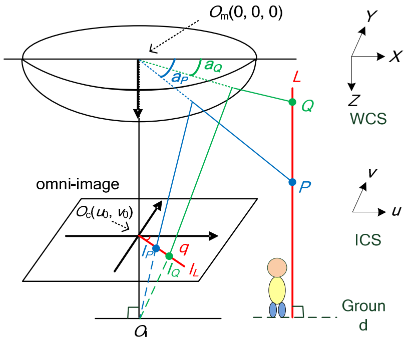
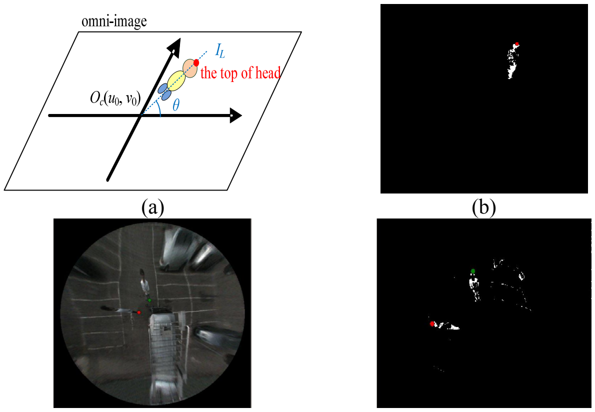


Figure 7. A vertical-line property of omni-camera: projection of a vertical line in real world will result in a line in omni-image going through the image center.



(a) (b)

(c) (d)

Figure 8. Passing-by person's head detection (a) Middle line of a person going through image center. (b) Result of one-person detection of Fig. 6(d) (head marked in red). (c) and (d) Result of two-person detection (feet and heads marked in red and green).

C. Computation of Passing-by Person's Location and Height

With the passing-by person's head location detected in the upper and lower images taken by an omni-imaging device, the formulas derived in Section II.C may be used to compute the location (X, Y) of the person and his/her body height h. Specifically, the coordinates (X, Y) are just those described by Eqs. (11) and (12). And as can be figured out easily from Figs. 4 and 7, the person's body height h may be computed as:

$$h = d_{\mathrm{h}} - Z \tag{13}$$

where d_{h} is the height of the upper camera with respect to the ground and Z is as that computed by Eq. (9) because P is the passing-by person's head point (shown as the red point in Fig. 8(b)). Note that the origin of the WCS is located at the mirror base center of the upper omni-camera. Also, by a back projection of the person's feet location at (X, Y, d_{h}) in the WCS into the omni-image, the person's feet can be marked for easier observation (e.g., see the color points in Fig. 8(c)).

Detection of passing-by persons and computation of their location and height features as described previously is based on the technique of background subtraction, which, though requiring background image updating whenever necessary, still has many applications with a video surveillance vehicle as the working platform, like car driver identification, alert-region protection, car stealing prevention, passing-by person counting, violent attack warming, etc.

IV. Generation of Top- and Perspective-view Images

A. Construction of Top-view Images Using Upper Cameras

Because the field of view (FOV) of the upper camera in an imaging device is wider than the lower one, we use the upper camera to construct a top-view image of the video surveillance vehicle's surrounding area. The height d_{h} of the camera affixed on the vehicle's roof is known in advance by manual measurement. Also, for simplicity it is assumed that all pixels around the vehicle in the image are projections of real-world points *on the ground*, as illustrated in Fig. 9.

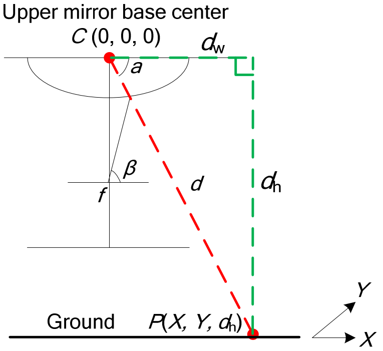


Figure 9. Ray tracing of a real-world point P on ground onto the mirror.

We adopt a backward-mapping scheme based on the use of the pano-mapping table to compute the desired top-view image. First, according to Fig. 9 we may compute the horizontal distance, d_{w}, between a ground point P and the mirror base center C in terms of the coordinates (X, Y, d_{h}) of P in the WCS as follows:

$$d_{\mathrm{w}} = \sqrt{X^2 + Y^2}\,. \tag{14}$$

Accordingly, the azimuth angle θ of P can be derived as:

$$\theta = \cos^{-1}(X/d_w) = \sin^{-1}(Y/d_w). \quad (15)$$

On the other hand, because $d_w = d_h \times \cot\alpha$ where d_h is the height of the upper camera's mirror base, we can compute the elevation angle α of P by:

$$\alpha = \tan^{-1}(d_h/d_w). \quad (16)$$

Furthermore, using the radial stretching function $r = f_s(\alpha)$ mentioned previously, the radial distance r corresponding to α can be derived. Accordingly, by the rotational-invariance property of omni-imaging, the coordinates (u, v) of the image pixel p corresponding to P can be obtained from (15) as:

$$u = r\cos\theta;\ v = r\sin\theta. \quad (17)$$

As a result, a complete top-view image can be computed. A top-view image of a parking area so computed with the surveillance vehicle in the middle is shown in Fig. 10. Note that all the cars appearing in the figure are distorted, and this phenomenon comes from their violation of the assumption that the processed points are projections of real-world points *on the ground*, as mentioned previously. However, the result is still good for visual inspection of the vehicle's surround.

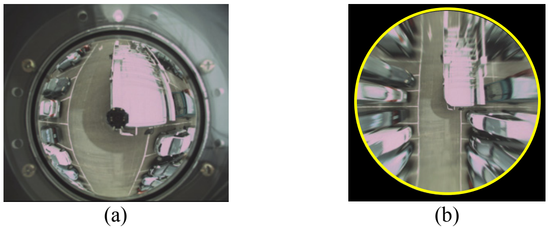


Figure 10. An omni-image and its corresponding top-view images. (a) Omni-image. (b) Top-view image obtained from backward mapping.

B. *Merging of two top-view images into a single one*

By the way describe above, we can construct two top-view images I_{t1} and I_{t2} using the omni-images taken from the two upper cameras, like those shown in Figs. 11(a) and 11(b). It is assumed that the relative position of the two upper cameras on the vehicle roof are measured manually in advance and described by a pair of offsets (C_W, C_L) with C_W and C_L being the horizontal and vertical distances between the two imaging devices, respectively. It is desired to merge the two top-view images I_{t1} and I_{t2} into a single one, denoted as I_{int}, to get a wider FOV of the vehicle's surround. To do this, we divide first the FOV covered by the desired top-view image I_{int} into two parts: one from the upper half of I_{t1}, and the other from the lower half of I_{t2}. All the pixels in the latter part then are shifted for the offset (C_W, C_L) before being merged into the former. An example of a top-view image so obtained is shown in Fig. 11(c). To obtain a *better-looking* top-view image, an elliptical shape is used further as a *viewing window*, and all the pixels outside the window are discarded, yielding an elliptical-shaped image. For Fig. 11(c), the result of this process is shown in Fig. 11(d).

C. *Superimposition of Surveillance Vehicle Shape on Top-view Image*

Because the imaging devices are affixed on the vehicle's roof, the vehicle shape in the top-view images always appears at *fixed* locations in the taken omni-images. Therefore, it is feasible to superimpose a *real surveillance vehicle shape* S_{rv} on the elliptical-shaped top-view image like that in Fig. 11(d) as the *central landmark* of the image, making observation of the vehicle's position in the surround more convenient. For this, the following steps are conducted: (1) trace and segment out manually the real top-view shapes S_1 and S_2 of the surveillance vehicle in the two top-view images $I_{t1}{}^o$ and $I_{t2}{}^o$, respectively in the learning phase; (2) superimpose S_1 and S_2 respectively on I_{t1} and I_{t2} acquired in each patrolling phase to segment out by image matching, and mark in yellow the surveillance vehicle's shape area A in I_{t1} and I_{t2} before they are merged into one, resulting in a figure like that shown in Fig. 12(a); (3) apply a texture synthesis scheme to fill ground texture into the marked surveillance vehicle shape area A; and (4) superimpose a pre-segmented real top-view vehicle shape S_{rv} like that shown in Fig. 12(b) on the resulting image of the last step, yielding a *better-looking* output image like that shown in Fig. 12(c). Note that S_{rv} is obtained in the learning phase by manual segmentation from a top-view image like that of Fig. 11(a).

Step (3) above is carried out by the following way: for each pixel p_c in the surveillance vehicle shape A, find the pixel p_n which is outside A and closest to p_c, and use the color of p_n to fill up p_c. Also, the relations of pixels in A with those outside A are kept in a table for use in the later top-view image generation process to improve the top-view image display speed.

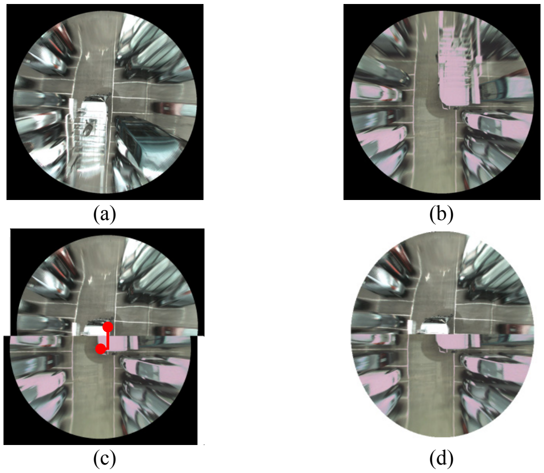


Figure 11. Top-view image construction. (a) Right-front top-view image. (b) Left-rear top-view image. (c) Integrated top-view image (red dots: centers of original images). (d) Top-view image viewed in an elliptical shape.

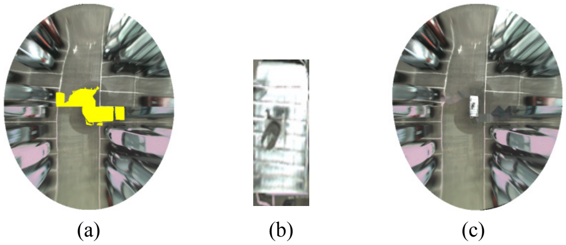


Figure 12. Superimposing real surveillance vehicle shape onto top-view image. (a) Top-view image with vehicle shape marked in yellow. (b) Pre-segmentd real shape of video surveillance vehicle. (c) Result of ground texture filling and superimposition of real surveillance vehicle shape.

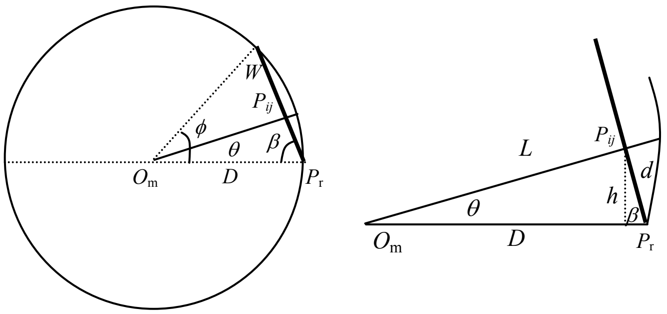
D. *Generation of Perspective-view Images*

The construction of a perspective-view image I_P from an omni-image I_o is based on the use of the pano-mapping table [12]. The major steps include: (1) map each pixel p in the desired perspective-image I_P at coordinates (k, l) to a pair of elevation and azimuth angles (α, θ) in the pano-mapping table according to the geometry of the desired perspective transformation; (2) find the image coordinates (u, v) in the table entry corresponding to (α, θ); and (3) assign the color channel values of pixel p in I_P to be those of the pixel at coordinates (u, v) in I_o. The detail of mapping (k, l) to (α, θ) in Step (1) is described as follows.

Assume that the perspective-view image I_p we want to generate from I_o is at a distance D to the mirror base center O_m and has $M_p \times N_p$ pixels. Also assume that I_p is the image of a planar rectangular region A_P with size $W \times H$ and is perpendicular to the floor in the real-world space, as illustrated from the top view by Fig. 13.

(a) Computing the azimuth angle θ —

The angle ϕ spanned by the width W of I_p as shown in Fig. 13(a) may be derived by the law of cosines to satisfy the equality $W^2 = D^2 + D^2 - 2 \times D \times D \times \cos\phi$ so that ϕ may be computed as $\phi = \cos^{-1}[1 - W^2/(2D^2)]$. Also, the angle β in the figure is just $\beta = (\pi - \phi)/2$. Next, let P_{ij} denote the intersection point of the light ray R_P projected onto the image pixel p and the planar region A_P mentioned previously. Then, we may compute the distance d between point P_{ij} and the border point P_r shown in Fig. 13(b) by linear proportionality to be $d = k \times W/M_P$ because A_P has a width of W, I_p has a width of M_p pixels, and pixel p has the coordinate k in the horizontal direction.



(a) (b)

Figure 13. Illustration of top view of a generated perspective-view image.

Furthermore, by the law of cosines again the distance L between point P_{ij} and the mirror base center O_m as shown in Fig. 13(b) satisfies the equality $L^2 = D^2 + d^2 - 2 \times d \times D \times \cos\beta$. Also, the distance h from point P_{ij} to the line segment O_mP_r may be computed as $h = d \times \sin\beta$. Finally, the azimuth angle θ of point P_{ij} with respect to O_mP_r satisfies $\sin\theta = h/L$ which, by the equalities derived above, leads to the following desired value:

$$\theta = \sin^{-1}(h/L) = \sin^{-1}[\frac{d \times \sin\beta}{\sqrt{D^2 + d^2 - 2 \times d \times D \times \cos\beta}}]. \qquad (18)$$

(b) Computing the elevation angle —

An illustration of the involved imaging configuration for computing the elevation angle α from a lateral view is shown in Fig. 14. The height of region A_P is H and image I_p is divided into N_p intervals in the vertical direction. And so, by linear proportionality again, the height of P_{ij} may be computed to be $H_p = (l \times H)/N_p$ where l is the coordinate of pixel p in the vertical direction. Finally, by trigonometry, the desired elevation angle α may be derived to be:

$$\alpha = \tan^{-1}(H_p/L). \qquad (19)$$

This completes the derivations of the azimuth-elevation angle pair (θ, α) conducted in Step (1) mentioned above. Note that in these derivations, the start direction (specified by the line segment O_mP_r) of the angle ϕ spanned by the width W of I_p, as shown in Fig. 14, coincides with the horizontal direction 0°, resulting in the azimuth angle θ. Of course, perspective-view images for other azimuth angles may also be generated. A convenient scheme to do this is described next.

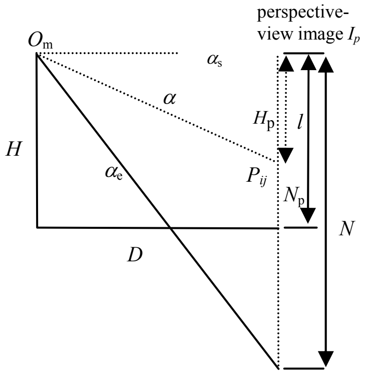


Figure 14. Lateral configuration for generating a perspective-view image.

E. Generation of Perspective-view Images Specified with Mouse Clicks or Panel Touch

In this study, a technique is proposed to enable a user to change the view direction of the perspective-view image conveniently by mouse clicking, or equivalently, by panel touching). Fig. 15(a) shows a perspective-view image generated from the omni-image shown in Fig. 15(b). According to the previous derivations and an observation of the images in the figure, we can see that there exists a relation between the mouse motion direction and the viewing direction of the generated perspective-view image. Specifically, the horizontal motion of a mouse specified by its horizontal location M_x may be used to define the azimuth angle θ of the space point P_c corresponding to the image center p_c; and similarly the vertical motion of the mouse specified by its vertical location M_y may be used to define the elevation angle α of P_c.

Accordingly, as an interaction with the proposed system via mouse clicking on the computer screen which shows a perspective-view image like that of Fig. 15(a), we define in this study an angle value θ_{mouse} according to the mouse click location in the horizontal direction so that θ_{mouse} becomes larger and larger as the mouse moves from right to left in the image of Fig. 15(a); and then modify the equation for computing the azimuth angle, namely, Eq. (18), derived previously to be:

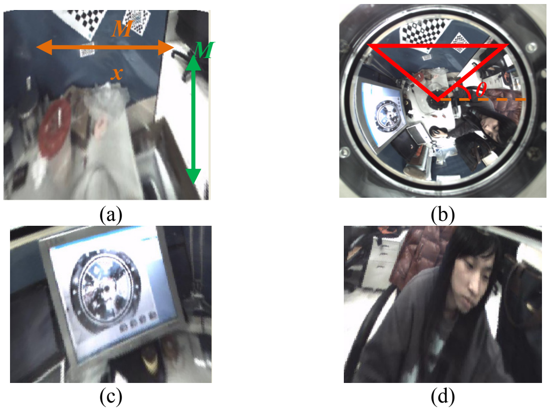
$$\theta = \sin^{-1}(h/L) + \theta_{mouse}. \qquad (20)$$

Similarly, we define another angle value α_{mouse} according to the mouse click location in the vertical direction such that α_{mouse}

increases gradually as the mouse moves from top to bottom in Fig. 15(a); and then modify Eq. (19) to be:

$$\alpha = \tan^{-1}(H_p/L) + \alpha_{\text{mouse}}. \qquad (21)$$

The user interface becomes friendlier after adding the two variables θ_{mouse} and α_{mouse}, and a user of the surveillance vehicle can now choose any view direction conveniently by mouse clicking (or panel touching) to observe the corresponding perspective-view image of a scene of his/her interest. Two more experimental results of perspective-view images generated in this way are shown in Figs. 15(c) and 15(d).



(a) (b) (c) (d)

Figure 15. Corresponding omni-image and perspective-view image. (a) A perspective-view image. (b) Omni-image part (enclosed roughly by the red triangle) from which (a) was generated. (c) and (d) Two other perspective-view images generated from (b) by mouse clicking on (a).

V. Experimental Results and Discussions

A. Experimental Results of 3D Data Computation

At first, we conducted an experiment to test the precision of the computed 3D data by computing the positions of some real-world points and compared them with those obtained manually. We laid an omni-imaging device on the ground and took images of a calibration pattern with alternating black and white grids and some black cross shapes as shown in Fig. 16. The width between every two consecutive grids on the pattern is 5cm, and the bigger width between every two crosses is 25 cm. We picked out 15 pairs of corresponding pixels from these grids and shapes in the two taken images. An example is shown in Fig. 17. We then calculated, by Eqs. (9) in Section II.C, the horizontal distance d_w and the height Z of the real-world point corresponding to each pair of corresponding image pixels. In addition, to increase the computation accuracy, we used *four* radial stretching functions instead of just one in the pano-mapping process, each function dealing with a quarter of the omni-image, as shown in Fig. 18. This measure is especially necessary when the assumption of radial symmetry of the mirror surface mentioned previously is *not* valid, which happens to be the case encountered in this study because of imperfect manufacturing of the mirrors.

The results of calculations of the 3D data, d_w and Z, of 15 selected landmark points on the calibration pattern using the taken images are shown in Tables 2 and 3, which resulted respectively from the use of the pano-mapping tables of Tables A and B depicted in Fig. 18. As can be seen, the differences between the measured data and the computed ones are all small, resulting in small RMSE values (each defined to be the square root of the average of all the difference values). All the error rates (each defined to be the ratio of the RMSE over the average measured data value) can be seen to be about 5%, which are good for practical applications. We also computed the values of the correlation coefficient between the difference values and the azimuth/elevation angles, respectively, to see the affections of such angles to the 3D computation results. The computed correlation coefficient values show that only the variations of the elevation angle have medium influences (with correlation coefficient values of −0.48 and −0.41) on the precisions of the computed location values (d_w) of the real-world points.

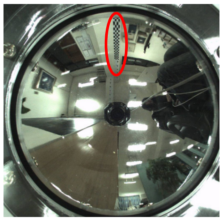
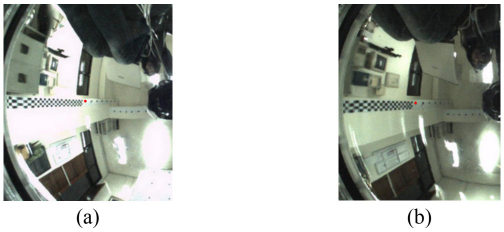


Figure 16. Omni-image of a calibration pattern with black and white grids and black cross shapes on a wall taken by an omni-camera laid on the ground.



(a) (b)

Figure 17. Illustration of picked out pairs of corresponding pixels in two omni-images (marked by two red dots).

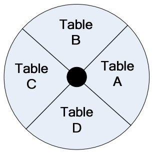


Figure 18. Four image regions corresponding to four radial stretching functions used in constructing pano-mapping tables.

Table 2. Statistics of computed 3D data using pano-mapping table Table A.

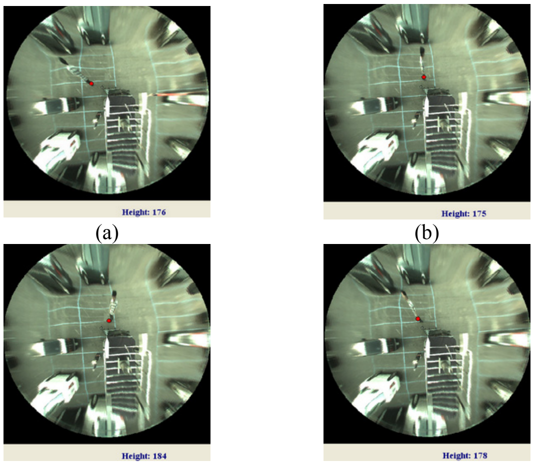
Point No.	Azimuth angle (º)	Elevation angle (º)	(1) Measured 3D real-world point data (cm)		(2) Computed 3D real-world point data (cm)		(3) Differences (1) – (2) (cm)	
	θ	α	d_w	Z	d_w	Z	d_w	Z
1	0.00	54.95	249	355	250.2	355.4	-1.20	-0.40
2	0.00	52.96	249	330	251.4	331.4	-2.40	-1.40
3	0.00	50.77	249	305	255	310.2	-6.00	-5.20
4	0.00	48.35	249	280	260.6	290.7	-11.60	-10.70
5	0.00	45.68	249	255	247.6	253.6	1.40	1.40
6	0.00	42.73	249	230	257.7	237.2	-8.70	-7.20
7	0.00	39.46	249	205	236.6	197	12.40	8.00
8	0.00	37.35	249	190	227.7	175	21.30	15.00
9	0.00	34.32	249	170	223	154.1	26.00	15.90
10	0.00	31.07	249	150	231.8	140.5	17.20	9.50
11	0.00	27.57	249	130	238	124.2	11.00	5.80
12	10.69	54.45	227.96	319	219.5	300.2	8.46	18.80
13	12.00	57.47	203.45	319	185.5	286.7	17.95	32.30
14	15.85	64.10	154.89	319	156.1	323.3	-1.21	-4.30
15	18.84	67.67	131.02	319	130.5	321.2	0.51	-2.20
Average			230.42	258.4				
RMSE							12.52	12.336
Error rate = RMSE / Average							5.43%	4.77%
Correlation coefficient	Azimuth angles						-0.03	0.17
	Elevation angle						-0.48	-0.18

Table 3. Statistics of computed 3D data using pano-mapping table Table B.

Point No.	Azimuth angle (°)	Elevation angle (°)	(1) Measured 3D space point data (cm)		(2) Computed 3D space point data (cm)		(3) Differences (1) – (2) (cm)	
	θ	α	d_w	Z	d_w	Z	d_w	Z
1	90.00	54.95	249	355	249	355.2	0.00	-0.20
2	90.00	52.96	249	330	251.4	331.4	-2.40	-1.40
3	90.00	50.77	249	305	255	310.2	-6.00	-5.20
4	90.00	48.35	249	280	260.6	290.7	-11.60	-10.70
5	90.00	45.68	249	255	247.6	253.6	1.40	1.40
6	90.00	42.73	249	230	257.7	237.2	-8.70	-7.20
7	90.00	39.46	249	205	236.6	197	12.40	8.00
8	90.00	37.35	249	190	231.3	178.4	17.70	11.60
9	90.00	34.32	249	170	232.8	160.8	16.20	9.20
10	90.00	31.07	249	150	237.3	143.2	11.70	6.80
11	90.00	27.57	249	130	238	124.2	11.00	5.80
12	100.69	54.45	227.96	319	219.7	303.9	8.26	15.10
13	102.00	57.47	203.45	319	185.5	286.7	17.95	32.30
14	105.85	64.10	154.89	319	156.1	323.3	-1.21	-4.30
15	108.84	67.67	131.02	319	130.5	321.2	0.52	-2.20
Average			230.42	258.4				
RMSE							10.436	11.121
Error rate = RMSE / Average							4.53%	4.30%
Correlation coefficient	Azimuth angles						0.04	0.21
	Elevation angle						-0.41	-0.12

B. Experimental Results of passing-by person detection

We have also tested the technique we proposed for passing-by person detection and localization. The environment for this experiment is an open space in a parking area. Because of the property of imaging projection, after the region of a passing-by person is found in the image, the body point which is farthest to the center of the image is located as the position of the person's head, as mentioned previously and shown in Figs. 6 and 8. In this experiment, we processed an image sequence with a person walking around the video surveillance vehicle, and the person was detected successfully in the image frames. A sequence of top-view images was constructed and the locations of the person were correctly computed. Some results are shown in Fig. 19 in which the red points are used to mark the detected person's feet positions.



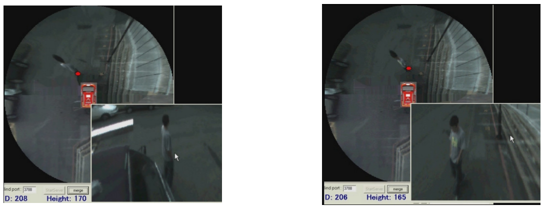
(a) (b)

(c) (d)

Figure 19. Examples of passing-by person detection. (a)~(d) Detection results shown as top-view images with red points used to mark detected person's feet.

C. Experimental Results of Integration of System Functions

A system integrating the proposed techniques has been implemented, including the functions of (1) construction of a top-view image of the surveillance vehicle's surrounding area; (2) detection of passing-by persons around the vehicle; (3) generation of a perspective-view image whose view direction may be specified by mouse clicking or is determined automatically to show the detected person. Each camera of model Artcam-200MI used in the proposed system takes 0.1 sec. to acquire a 1600×1200 image, and the processing cycle time of the system is 0.2 sec. (including person detection and top- and perspective-view image generation). The height of the lower cameras, when affixed to the surveillance vehicle, is 231.8 cm and that of the higher ones is 256 cm. Two examples of the experimental results using the system are shown in Fig. 20.



(a) (b)

Figure 20. Examples of generated images with right lower corner in each image showing generated perspective-view image of a detected person.

D. Further Survey of Existing methods and Comparisons

(a) About catadioptric optics — Baker and Nayar [15] derived the class of single-lens single-mirror catadioptric sensors that have single viewpoints. Geyer and Daniilidis [16] proposed a unifying model for the projective geometry induced by catadioptric sensors. While good theories can be found in these papers, we tried in this study to design an omni-camera with suitable parameters to *fit* the roof structure of a vehicle used in this study.

(b) About catadioptric stereo — Xiong, et al. [17] described various ways of designing omni-directional stereo vision systems and related techniques for image unwarping, stereo matching, and moving object detection. Su et al. [18] obtained obstacle information omni-directionally by an omni-directional stereo vision system with a perspective camera coupled with two hyperbolic mirrors. Differently, the method proposed in this study combines in a *novel* way the pano-mapping technique [12] and the use of a two-camera omni-imaging device for 3D data computation and fast top- and perspective-view image generation.

(c) About object detection on moving vehicles — Gandhi and Trivedi [3] used images of two omni-cameras affixed to car side mirrors to synthesize perspective-view images, and detect in-front vehicles by binocular stereo and lateral ones by motion stereo. They also used parametric ego-motion compensation for an omni-directional vision sensor to detect surrounding events [19] [20]. The proposed method instead emphasizes 3D detection of both passing-by persons' locations and heights, using omni-images *directly* without generating perspective views to speed up the detection process. This is made possible by the uses of the designed two-camera omni-imaging devices and the *single* pano-mapping technique.

(d) About top-view image generation — Ehlgen and Pajdla [21][22] installed two omni-cameras on the side mirrors of a truck to generate a bird's-eye-view image covering the *frontal* scene. The top-view image generated in this study instead

covers the *entire* vehicle surrounding. Liu, et al. [23] affixed six fisheye cameras around a car to generate a bird's-eye-view image of the surround for driving assistance. No 3D data computation was considered, contrasting with our method which computes 3D feature point data.

(e) About human detection — Liu et al. [24] detected human motion with an omni-camera on a mobile robot using ego-motion compensation and temporal differencing *after* unwarping omni-images. The proposed method of this study instead detects passing-by persons *directly* from omni-images. Ng, et al. [25] used multiple omni-vision sensors to synthesize perspective views and track human activities using N-ocular stereo techniques without acquiring range information. In contrast, our method detects human beings in vehicle surroundings and computes their 3D features (location and height) for various applications.

(f) About perspective-view image generation — Huang et al. [26] presented an in-car omni-imaging system which processes acquired videos to obtain *direction-fixed* virtual perspective views on the driver, passengers, and frontal scenes using pan, tilt, and zoom parameters, in contrast with our method which generates *direction-changing* perspective views on passing-by persons without using camera parameters. Ng, et al. [25] generated images of both a walking person's view and an observer's view using a *range-space search technique*, while our method uses the *pano-mapping approach*. Kawasaki et al. [27] obtained 3D information by spatio- temporal analysis of omni-images using calibrated camera parameters, contrasting again with the tabular pano-mapping technique used by our method, involving no camera parameter.

(g) Comparisons about implemented functions — The result of a more detailed comparison of the previously-mentioned methods (except [17] which is a survey paper) with the proposed one in terms of a set of implemented functions is listed in Table 4, from which it can be seen that the proposed method has integrated more functions than the others.

VI. Conclusions

A pair of two-camera omni-imaging devices has been designed properly for use on the top of a video surveillance vehicle to monitor passing-by persons. The two devices are affixed to the right-front and left-rear of the vehicle roof efficiently to facilitate generation of a top-view image which covers the vehicle surrounding area. A new 3D data computation technique based on the pano-mapping concept and the rotational invariance property of the omni-image has been proposed. Because of the use of table lookup and analytic computation formulas, the technique can be implemented to satisfy real-time applications. A passing-by person appearing around the surveillance vehicle can be detected automatically using an omni-image property about upright objects, with his/her location and body height computed by the proposed 3D data acquisition technique. A top-view image of the vehicle's surrounding area with a real vehicle shape inserted properly in the middle is generated by registering two omni-images taken by two upper cameras. Perspective-view images covering the detected person or any interesting scene spot can also be generated in real-time automatically or by mouse clicking for convenient inspection. The system is useful for many security monitoring applications around the video surveillance vehicle. The experimental results show the feasibility and precision of the proposed method for practical applications.

References

[1] C. Regazzoni, V. Ramesh and G. Foresti, "Special issue on video communications, processing, and understanding for third generation surveillance systems," *Proc. of IEEE*, vol. 89, no. 10, pp. 1355–1539, Oct. 2001.

[2] W. Hu, T. Tan, L. Wang and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334-352, Aug. 2004.

[3] T. Gandhi and M. M. Trivedi, "Vehicle surround capture: survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 3, Sept. 2006, pp. 293-308.

[4] C. Micheloni, G. L. Foresti, C. Piciarelli and L. Cinque, "An autonomous vehicle for video surveillance of indoor environments," *IEEE Trans. on Vehicular Technology*, vol. 56, no. 2, pp. 487-498, March 2007.

[5] K. C. Chen and W. H. Tsai, "Guidance of indoor vision-based autonomous vehicles for security patrolling by a SIFT-based vehicle localization technique using along-path object information," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 7, Sept. 2010, pp. 3261-3271.

[6] Y. Onoe, N. Yokoya, K Yamazawa, and H. Takemura, "Visual surveillance and monitoring system using an omnidirectional video camera," *Proc. of 1998 Int'l Conf. Pattern Recog.*, Vol. 1, pp.588-592, Brisbane, Australia, Aug. 16-20, 1998.

[7] T. Mituyosi, Y. Yagi, and M. Yachida, "Real-time human feature acquisition and human tracking by omnidirectional image sensor," *Proc. of IEEE Int'l Conf. on Multisensor Fusion & Integration for Intelligent Systems*, pp. 258-263, Tokyo, Japan, July 30-Aug. 1, 2003.

[8] S. Morita, K. Yamazawa, and N. Yokoya, "Networked video surveillance using multiple omnidirectional cameras," *Proc. of 2003 IEEE Int'l Symp.on Computational Intelligence in Robotics & Automation*, vol. 3, pp. 1245-1250, Kobe, Japan, July 16-20, 2003.

[9] H. Koyasu, J. Miura and Y. Shirai, "Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments," *Proc. of 2001 IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, vol. 1, pp. 31-36, Maui, Hawaii, USA, Oct. - Nov., 2001.

[10] H. Ukida, N. Yamato, Y Tanimoto, T Sano, and H. Yamamoto, "Omni-directional 3D measurement by hyperbolic mirror cameras and pattern projection," *Proc. of 2008 IEEE Conf. on Instrumentation & Measurement Technology*, pp. 365-370, Victoria, BC, Canada, 2008.

[11] J. I. Meguro, J, I, Takiguchi, and Y, A, T, Hashizume, "3D reconstruction using multi-baseline omni-directional motion stereo based on GPS/DR compound navigation system," *Int'l J. of Robotics Research*, vol. 26, no. 6, pp.625-636, June 2007.

[12] 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 Proc.*, vol. 1, no. 2, pp. 149-155, June 2007.

[13] W. H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, & Image Proc.*, vol. 29, pp. 377-393, 1985.

[14] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Prentice-Hall: Englewood Cliffs, NJ, USA, 2002.

[15] S. Baker and S. Nayar, "A theory of single-viewpoint catadioptric image formation," *Int'l J. of Computer Vision*, vol. 35, no. 2, 1999, pp. 175-196.

[16] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *Int'l J. of Computer Vision*, vol. 45, no. 3, 2001, pp. 223-243.

[17] Z. Xiong, W. Chen and M. J. Zhang, "Catadioptric omni-directional stereo vision and its applications in moving objects detection," in *Computer Vision*, Z. Xiong, ed., InTech: Rijeka, Croatia, Nov. 2008, pp. 493-518.

[18] L. C. Su, C. J. Luo and F. Zhu, "Obtaining obstacle information by an omnidirectional stereo vision system," *IEEE Int'l Conf. on Information Acquisition*, Aug. 20-23, 2006, Weihai, Shandong, China, pp. 48-52.

[19] T. Gandhi and M. M. Trivedi, "Motion analysis for event detection and tracking with a mobile omni-directional camera," *Multimedia Systems Journal*, vol. 10, no. 2, 2004, pp. 131-143.

Table 4. Comparison of existing methods (O: yes, X: no; ***: using pano-mapping table).

	omni-image unwarping	Top view generation	Integration of multiple images	Real-time processing	Applied in outdoor environment	Passing-by people detection	Object distance estimation	Object height estimation
Our method	O	O	O	O	O	O	O	O
Su et al. [18]	O	X	X	O	X	X	O	X
Gandhi & Trivedi [3]	O	X	X	O	O	X	O	X
Gandhi & Trivedi [20]	X	O	X	O	O	X	O	X
Ehlgen & Pajdla [21, 22]	X	O	O	O	O	X	X	X
Liu, et al. [23]	O	O	O	O	O	X	X	X
Liu et al. [24]	O	X	X	O	X	O	X	X
Ng, et al. [25]	O	X	X	O	X	O	O	X
Huang et al. [26]	O	X	X	O	O	O	X	X
Kawasaki et al. [27]	O	X	X	X	O	X	O	O

[20] T. Gandhi and M. Trivedi., "Parametric ego-motion estimation for vehicle surround analysis using an omnidirection camera," *Machine Vision and Applications*, vol. 16, 2005, pp. 85-95.

[21] T. Ehlgen and T. Pajdla, "Maneuvering aid for large vehicle using omnidirectional cameras," *Proc. of 8th IEEE Workshop on Applications in Computer Vision,* Austin, TX, USA, Feb. 2007, p. 17

[22] T. Ehlgen and T. Pajdla, "Monitoring surrounding areas of truck-trailer combinations," *Proc. of 5th Int'l Conf. on Computer Vision Sys*tems, Bielefeld, Germany, March 2007, pp. 21-24.

[23] Y. C. Liu, K. Y. Lin, and Y. S. Chen, "Bird's-eye view vision system for vehicle surrounding monitoring," *Proc. of Conf. on Robot Vision*, Berlin, Germany, Feb., 2008, pp. 207-218.

[24] H. Liu, N. Dong and H. Zha, "Omni-directional vision based human motion detection for autonomous mobile robots," *Proc. of IEEE Int'l Conf. on Systems, Man & Cybernetics*, Oct. 2005, pp. 2236-2241.

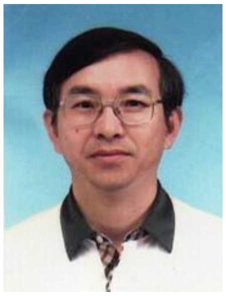
[25] K. C. Ng, H. Ishiguro, M. Trivedi, and T. Sogo, "An integrated surveillance system — human tracking and view synthesis using multiple omni-directional vision sensors," *Image and Vision Computing*, vol. 22, pp. 551–561, 2004.

[26] K. S. Huang, M. M. Trivedi and T. Gandhi, "Driver's view and vehicle surround estimation using omnidirectional video stream," *Proc. of IEEE Intelligent Vehicles Symp.*, Columbus, Ohio, USA, June 2003, pp. 444-449.

[27] H. Kawasaki, K. Ikeuchi and M. Sakauchi, "Spatio-temporal analysis of omni-images," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, June 2000, pp. 577-584.

Pei-Hsuan Yuan received the B.S. degree in computer science and information engineering from National Chi Nan University, Taiwan in 2008 and the M.S. degree in computer science from National Chiao Tung University, Taiwan in 2010. She was a research assistant at the Computer Vision Laboratory in the Department of Computer Science at National Chiao Tung University from August 2008 to July 2010. She is now with Taiwan Semiconductor Manufacturing Co. in Taiwan. Her research interests include computer vision, image processing, video surveillance and applications.

Kuo-Feng Yang received the B.S. degree in mathematics from National Chung Hsing University, Taiwan in 2006 and the M.S. degree in computer science and information engineering from Yuan Ze University, Taiwan in 2008. He has been a research assistant at the Computer Vision Laboratory in the Department of Computer Science at National Chiao Tung University since August 2008. He is also working toward his Ph. D. degree there. His current research interests include image and vision computing, and autonomous vehicle applications.



Wen-Hsiang Tsai received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Department of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan, and from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taiwan.

Dr. Tsai has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has published 144 journal papers and 227 conference papers. Dr. Tsai has received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies.

Dr. Tsai's current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society in Taiwan and a Senior Member of the IEEE.