A New 3D Imaging System Using a Portable Two-camera Omni-imaging Device for Construction and Browsing of Human-reachable Environments[1]

Yu-Tung Kuo[1] and Wen-Hsiang Tsai[1, 2]

[1] Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan
[2] Department of Information Communication, Asia University, Taiwan

Abstract. As an improvement on existing street view systems which cannot be moved to certain environments to acquire images, a new 3D imaging system using a portable two-camera omni-imaging device is proposed. The device can be carried on one's back and moved on foot to any indoor and outdoor scene spot to acquire omni-images which cover the entire spherical view of the scene with an overlapping image band. By a space-mapping technique, the two omni-images are transformed into panoramic images, and the overlapping band is utilized to stitch them to form a single total panoramic image using a dynamic programming technique. Browsing of an environment with a series of scene spots can be conducted to see the perspective-view image of any spot in any direction by changing the viewpoint via the uses of four tools – the current perspective-view image, the total panoramic image, a walking path, and a three-view diagram. Experimental results show the feasibility of the system.

Keywords: 3D imaging system, street view, omni-image, scene spot.

1 Introduction

In recent years, many street view systems have been developed [1, 7, 8]. Fig. 1 shows two examples. To exploit a certain street scene, one may just browse a street view system to see the scene online without going there. This provides great value to applications like city touring, real estate sales, environmental management, etc. To collect street scenes, Google Street View Cars [2] were used (Fig. 2(a)). To enter narrow lanes, a Street View Trike was designed later (Fig. 2(b)). Also launched was a Street View Snowmobile for use on rough terrains (Fig. 2(c)). Aboard each of these vehicles is an imaging device with eight digital cameras, one fish-eye camera, and three laser range finders for scene collection. The device weighs about 150 kg.

(a)

(b)

Fig. 1. Examples of street view systems. (a) Street view of Google Map. (b) Street view of Bing Maps.

Though the use of the Street View Trek overcomes the incapability of the Street View Car in reaching narrow alleys, it is hard for a rider to pedal the heavy tricycle on steep ways. Also, although the Snowmobile has better mobility, it still cannot be

[1] This work was supported financially by the NSC project No. 98-2221-E-009-116-MY3.

ridden to some spaces like indoor stairways, mountains tracks, garden paths, etc. To reduce the system weight and widen the reachable environments, it is desired to design a new system which is portable by a person.

Specifically, as shown in Fig. 3, in this study we utilize only two omni-cameras to construct an imaging device which weights less than 10 kg. The device is affixed to a steel holder which can be carried conveniently by a person on his/her back and moved to anywhere on foot (Fig. 3(a)). The two omni-cameras are aligned coaxially in a back-to-back fashion (Fig. 3(b)), and can be used to acquire two omni-images simultaneously which cover the spherical surrounding space. To stabilize the imaging device during walking in order to avoid acquisition of vibrating images, a set of gimbals is used to hold the device (Fig. 3(c)).

(a) (b) (c)

Fig. 2. Google street view vehicles. (a) Street view Car. (b) Street View Trike. (c) Snowmobile.

(a) (b) (c)

Fig. 3. Proposed 3D imaging system. (a) The system on the back of a carrier. (b) The imaging device consisting of a pair of omni-cameras. (c) A set of gimbals used for stabilizing the cameras during walking.

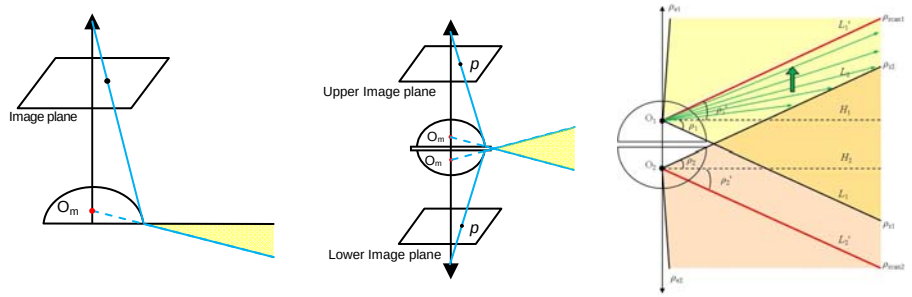
With environment reachability and imaging capabilities provided, there still arise many other technical issues which should be solved before the proposed system can be used for environment browsing, including: (1) transformation of the omni-images taken at a scene spot into panoramic ones; (2) stitching of the two panoramic images into a single one for use as an overview of the scene spot; (3) generation of a perspective-view image for inspection of the detail at the spot; (4) updating of the perspective-view image according to a chosen viewpoint; (5) provision of an interface for environment browsing, etc. All of these issues are solved in this study.

In the remainder of this paper, we describe the system configuration and processes in Sec. II, the adopted techniques for panoramic and perspective-view image generations in Sec. III, and the proposed technique for panoramic image stitching in Sec. IV, followed by experimental results in Sec. V and conclusions in Sec. VI.

2 System Configuration and Processes

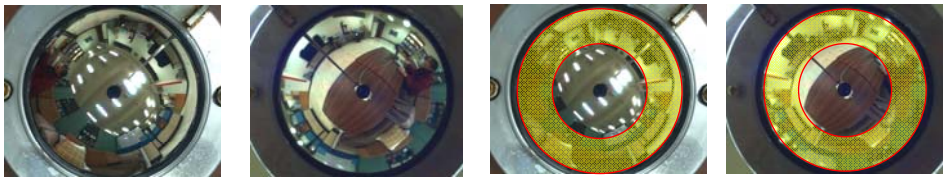
In this study, we align two catadioptric omni-cameras coaxially and back to back to form a new 3D image device as shown in Fig. 3(b) for acquiring omni-images at scene spots. Each omni-camera includes a projective camera and a hyperboloidal-

shaped mirror. The mirror bottom plane may be designed to go through a focal point of the hyperboloidal shape of the mirror. If two omni-cameras of this design are not connected seamlessly, a circular blind region not observable by either camera will appear undesirably. Therefore, in this study we design the mirror bottom plane to be *lower than* the focal point as shown in Fig. 4(a). This broadens the field of view (FOV) of the camera with an additional circular band as illustrated by the yellow portion in Fig 4(a). Moreover, after two omni-cameras with such mirrors are attached back to back, a circular overlapping band will appear in both omni-images as illustrated by the yellow region in Fig. 4(b). Fig. 5 shows an example of real omni-images taken by an omni-camera pair of such a design, in which the yellow portions in Figs. 5(c) and 5(d) show the circular overlapping band. This overlapping band is utilized skillfully in a new technique proposed in this study for image stitching (described later in Sec. IV).



(a) (b) (c)

Fig. 4. Illustration of proposed imaging device. (a) Light ray reflection with focal point higher than mirror bottom. (b) Combination of proposed omni-camera pair, creating an overlapping image band in both omni-images. (c) Overlapping of light rays reflected by mirrors of the two omni-cameras.



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

Fig. 5. Overlapping bands in upper and lower omni-images (a) Original upper omni-image. (b) Original lower omni-image. (c) Overlapping band in (a). (b) Overlapping band in (b).

The system processes include two stages: *environment learning* and *browsing*. The former includes three phases: (1) *system setup* – including calibration of the focal point location of the hyperboloidal shape, and establishment of a space-mapping table for each omni-camera for use in panoramic and perspective-view image generations; (2) *construction of panoramic images* – including image generation, inpainting [4], dehazing [5], blending [6], and stitching using a technique proposed in this study; (3) *database establishment* – recording relevant images, scene spot locations, and feature points in panoramic images. And the environment browsing stage includes two phases: (1) *display of the current scene spot view as a perspective-view image* – including generation of an initial perspective-view image and updating of it in

response to the user's view-changing operation; and (2) *display of the interface* – including generations of a panoramic image, a walking path, and a three-view diagram for scene spot browsing as well as updating of the interface.

3 Generations of Images of Various Views by Space Mapping

Generations of panoramic and perspective-view images in this study are based on the space-mapping approach proposed by Jeng and Tsai [3]. As illustrated in Fig. 6, each omni-image pixel p with coordinates (u, v) is the projection of a real-world point P with the light ray of the point going onto the mirror and reflected into the imaging plane, resulting in an azimuth angle θ and an elevation angle ρ whose values may be obtained by table-lookup using (u, v) according to a so-called *pano-mapping table* like that shown in Table 1.

To construct the table, with the mirror surface assumed radially symmetric, a mapping function $r = f(\rho)$ from ρ to the radial distance r of p with respect to the image center is set up first as $f(\rho) = a_0 + a_1\times\rho^1 + a_2\times\rho^2 + \ldots + a_5\times\rho^5$. Next, by 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 = 1~6, the coefficients a_0 through a_5 is obtained by computing the radial distances r_i for each P_i as $r_i = \sqrt{u_i^2 + v_i^2}$ and the elevation angle ρ_i for P_i as $\rho_i = \tan^{-1}(Z_i/\sqrt{X_i^2 + Y_i^2})$; and solving the six simultaneous equations $r_i = f(\rho_i) = a_0 + a_1\times\rho_i^1 + a_2\times\rho_i^2 + \ldots + a_5\times\rho_i^5$, where i = 1~6. Finally, for each real-world point P_{ij} with azimuth-elevation angle pair (θ_i, ρ_j), compute the image coordinates (u_{ij}, v_{ij}) of the corresponding pixel p_{ij} in the omni-image as $u_{ij} = r_j\times\cos\theta_i$; $v_{ij} = r_j\times\sin\theta_i$ where $r_j = f(\rho_j) = a_0 + a_1\times\rho_j^1 + a_2\rho_j^2 + \ldots + a_5\times\rho_j^5$ and fill them into the corresponding entry in the table (Table 1).

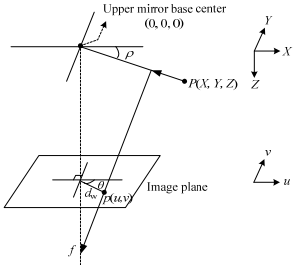


Fig. 6. An omni-camera with a mirror.

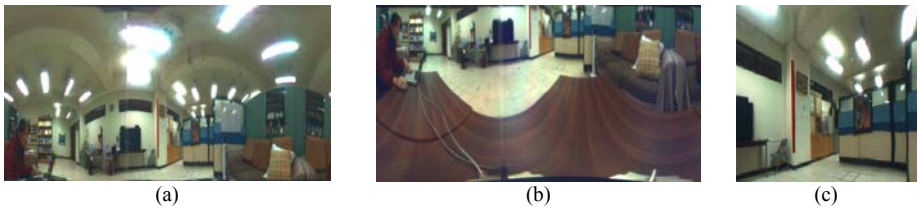
Table 1. An example of pano-mapping table of size $M\times N$.

	θ_1	θ_2	θ_3	θ_4	…	θ_M
ρ_1	(u_{11}, v_{11})	(u_{21}, v_{21})	(u_{31}, v_{31})	(u_{41}, v_{41})	…	(u_{M1}, v_{M1})
ρ_2	(u_{12}, v_{12})	(u_{22}, v_{22})	(u_{32}, v_{32})	(u_{42}, v_{42})	…	(u_{M2}, v_{M2})
ρ_3	(u_{13}, v_{13})	(u_{23}, v_{23})	(u_{33}, v_{33})	(u_{43}, v_{43})	…	(u_{M3}, v_{M3})
ρ_4	(u_{14}, v_{14})	(u_{24}, v_{24})	(u_{34}, v_{34})	(u_{44}, v_{44})	…	(u_{M4}, v_{M4})
...	…	…	…	…	…	…
ρ_N	(u_{1N}, v_{1N})	(u_{2N}, v_{2N})	(u_{3N}, v_{3N})	(u_{4N}, v_{4N})	…	(u_{MN}, v_{MN})

With the table so constructed, afterward whenever the image coordinates (u, v) of an pixel p in a given omni-image are known and checked by table lookup to be located in entry E_{ij} in Table 1, the corresponding azimuth-elevation angle pair $(\theta_i. \rho_j)$ in the table may be retrieved to describe the real-world point P corresponding to p.

With the pano-mapping table T for an omni-camera C available, we may construct a panoramic image I_q of the size of T from a given omni-image I_o taken by C by two steps: (1) for each entry E_{ij} in T with the azimuth and elevation angles (θ_i, ρ_j), take out the coordinates (u_{ij}, v_{ij}) filled in E_{ij}; and (2) assign the color values of pixel p_{ij} of I_o at coordinates (u_{ij}, v_{ij}) to pixel q_{ij} of I_q at coordinates (i, j). Also, we may construct a perspective-view image I_P from I_o for any viewpoint by three steps: (1) map each pixel p in I_P at coordinates (k, l) to a pair of elevation and azimuth angles (ρ, θ) in T

according to the top-view geometry of the desired perspective view [3]; (2) find by table-lookup the image coordinates (u, v) in T which correspond to (ρ, θ); and (3) assign the color values of the pixel at coordinates (u, v) in I_o to pixel p in I_P. For example, the panoramic images and a perspective-view image generated from the two omni-images shown in Fig. 5 are shown in Figs. 7(a) through 7(c).



(a) (b) (c)

Fig. 7. Images of different views generated from omni-images of Fig. 5. (a) Upper panoramic image. (b) Lower panoramic image. (c) Perspective-view image.

4 Stitching of Upper and Lower Panoramic Images

The new technique we propose for stitching two panoramic images of a scene spot into a *total panoramic image* includes the following steps.

(1) Localization of the overlapping band in the upper and lower omni-images —

As shown in Fig. 4(c), a partial FOV A_1 between L_1 and L_2 will appear in both omni-images as mentioned previously and it is desired to find out the range of elevation angles covering A_1 in either omni-image. The light ray (green in the figure) of each scene point on an boundary light ray L_2 of A_1 going through the mirror to the focal point O_1 is confined to be within a band B_1 formed by L_2 and L_1' where L_1' is the extreme light ray formed by an *infinitively far* scene point on L_2 and is so *parallel* to L_2. The elevation angles of the light rays within B_1 can be figured out to be within the range of $\rho_{upper} = \rho_1 + \rho_1'$ where ρ_1 is the range of the elevation angles formed by those of the boundary light ray L_1 and the horizontal line H_1 (the upper dotted line in the figure). The elevation angle of L_1 may be found by image processing from any point on the "circular image boundary" in the upper omni-image, and the elevation angle of H_1 is just 0^o. Therefore, ρ_1 may be computed explicitly, and by symmetry, so is ρ_2. Furthermore, since L_1' is parallel to L_2, ρ_1' may be figured out to be equal to ρ_2. Consequently, we have $\rho_{upper} = \rho_1 + \rho_1' = \rho_1 + \rho_2$ which is computable, and by symmetry so is the corresponding range ρ_{lower} for the lower omni-image, which actually just equals ρ_{upper}. Either of ρ_{upper} or ρ_{lower} finally may be transformed into an image band like the yellow circular region shown in Fig. 5(c) or 5(d).

(2) Selection of feature matching lines —

It seems feasible to stitch the two panoramic images by matching the overlapping image parts found in the last step. However, this does not work because the two image parts actually are *not* identical. One reason is that the two cameras take images at different heights so that the image scaling differs in the two images. Another reason is that radial-directional distortions exist on the mirrors of the two cameras so that different nonlinearities are created on the acquired images. Nevertheless, it is found in this study that the rotational invariance property of omni-imaging [3] can be used to solve this problem. The property says that image pixels of a certain radial direction in an omni-image are just the projections of some real-world points

appearing *in the same direction*. This in turn says that for each radial direction, there exist respectively in the omni-images two corresponding radial lines with *identical* image contents which are the projections of an identical set of real-world points.

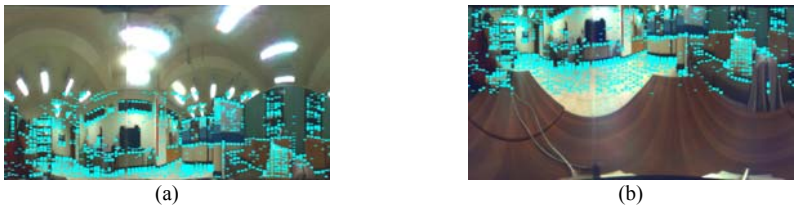
Accordingly, we propose in this study the idea of using radial lines to conduct image matching for stitching the upper and lower panoramic images. Such *radial matching lines* are selected manually in the learning stage to be of a set of 90 angular directions with equal intervals of 4^{o}. For each of such lines with angular direction θ, a pair of matching lines is set up, one in the upper omni-image and the other in the lower, both of the same direction θ. To facilitate manual pairings of such matching lines, as illustrated by Fig. 8, a cylindrical-shaped grid pattern with a series of 90 equally-distanced colored vertical lines was designed to wrap around each omni-camera, and an omni-image of the pattern was then acquired. From the two acquired images, the pairing work was conducted by visual inspection with the line locations recorded in the previously-mentioned database. Such a work need be conducted only once in the learning stage as long as the used omni-cameras are not changed.

(a) (b)

Fig. 8. Illustration of finding corresponding matching lines manually by visual inspection. (a) Grid pattern wrapped into a cylinder. (b) Corresponding matching lines in two omni-images.

(3) Feature point matching on matching lines

Stitching of the panoramic image pair proposed in this study starts from *feature point matching* within the overlapping band on each matching line pair. Feature points used are edge points detected on the matching lines in the grayscale versions of the panoramic images. For example, the feature point detection result with Fig. 7 as input is shown in Fig. 9. Due to differences in noise, distortion, and scaling between the two panoramic images, feature points appearing in the upper matching line might not appear in the lower, and vice versa, resulting in *insertions* and *deletions* of feature points in the matching lines. Regarding feature points which match mutually as *substitutions*, it is found in this study appropriate to use the *dynamic programming* technique to conduct *optimal* feature point matching on each matching line pair.



(a) (b)

Fig. 9. Feature detection result with Fig. 7 as input. (a) Result of processing upper panoramic image of Fig. 7(a). (b) Result of processing lower panoramic image of Fig. 7(b).

More specifically, assume that two radial matching lines respectively are represented by two feature point sequences $l_{\text{upper}} = < u_1, u_2, \ldots, u_m >$ and $l_{\text{lower}} = < l_1,$

l_2, …, l_n> where each u_i or l_j is a feature point. After transforming the RGB colors of the image points into corresponding HIS values (hue, intensity, and saturation values), each feature point w_i (u_i or l_i) is given an attribute which is the hue value at w_i, denoted as $H(w_i)$. Assume that the maximum hue value is $H_{\max}$. Each of the three previously-mentioned types of *edit* operations, insertion, deletion, and substitution, causes a *cost*, which is defined as: (1) *substitution cost* $C(u \to l) = |H(u) - H(l)|/H_{\max}$ for a feature point u with attribute $H(u)$ substituted by another l with feature $H(l)$; (2) *insertion cost* $C(\phi \to u) = 0.5$ for a feature point u being inserted with ϕ meaning *null*; or (3) *deletion cost* $C(l \to \phi) = 0.5$ for a feature point l being deleted. All the costs so defined are within the range of 0~1. Then, matching of feature points on a matching line pair is equivalent to finding the best edit sequence taking l_{upper} to l_{lower} according to the dynamic programming process, which we describe as follows, where D is an $m \times n$ table as illustrated in Fig. 10: (1) set $D(0, 0) = 0$; (2) for all $0 \le i \le m$ and $0 \le j \le n$, fill all unfilled $D(i, j)$ as the minimum of the following three values:

$V_{\text{i}} = D(i - 1, j) + C(\phi \to u)$ (for inserting u_i in l_{upper});

$V_{\text{s}} = D(i - 1, j - 1) + C(u_i \to l_j)$ (for substituting u_i in l_{upper} with l_j in l_{lower});

$V_{\text{d}} = D(i, j - 1) + C(l_j \to \phi)$ (for deleting l_j from l_{lower});

(3) with $D(m, n)$ being the total cost of the optimal editing sequence S taking l_{upper} to l_{lower}, trace the table back to obtain S for use in the subsequent steps.

(4) Use of knowledge to improve feature point matching

The attribute, average hue, used above is too simple to yield correct substitution results on all matching line pairs. So, additional knowledge about omni-imaging was exploited, yielding two constraints for use in dynamic programming so that unreasonable feature point matching results based on substitutions can be excluded.

The first constraint is about the consistency of the distance of the current matching point to former one: for a pair of feature points p_{u} and p_{l} on the upper and lower matching lines, respectively, to be valid, the distances d_{u} and d_{l} of them to their respective preceding matching feature points p_{u}' and p_{l}' found in the dynamic programming process should be roughly the same (because the upper and lower panoramic images, though different in content for reasons described previously, are still similar to a certain degree). The second constraint says that for two feature points be a matching pair, their respective elevation angles can differ only for a tolerable range R_{e} as can be verified from Fig. 11. Specifically, if L_{up} is the light ray of a feature point P observable by the upper camera, then in the extreme case that P is located infinitively far away, the corresponding light ray going into the lower camera is just L_{low} which is *parallel* to L_{up} so that the elevation angles ρ_{low} and ρ_{up} are equal. Accordingly, the range of elevation angles for P to be observable by both cameras can be figured out to be in the range from ρ_{s2} to $\rho_{s2} + \rho_2 - \rho_{\text{low}} = \rho_{s2} + \rho_2 - \rho_{\text{up}}$ which is computable because ρ_{s2} is the elevation angle of L_2, ρ_{up} is the elevation angle of point P, and ρ_2 is as shown in Fig. 4(c). This range can be used as R_{e} mentioned above.

These two constraints are applied in Step (2) of the dynamic programming process: only when both constraints are satisfied can a substitution operation be considered for a feature point pair; otherwise, only the insertion and deletion operations are used. The result of applying this revised dynamic programming process to Fig. 9 is shown in Fig. 12, where the red points represent the matching pixels obtained by substitutions and the blue ones mean those obtained by insertions and deletions.

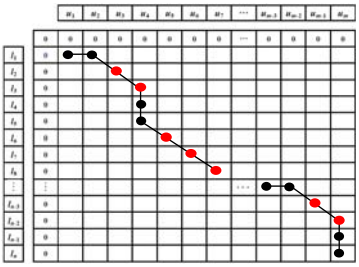


Fig. 10. Illustration of matching result by dynamic programming.

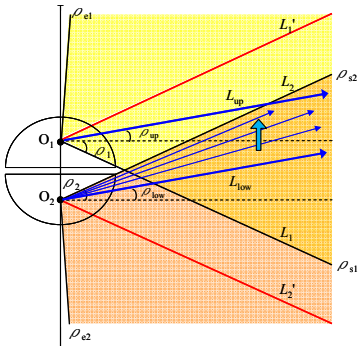
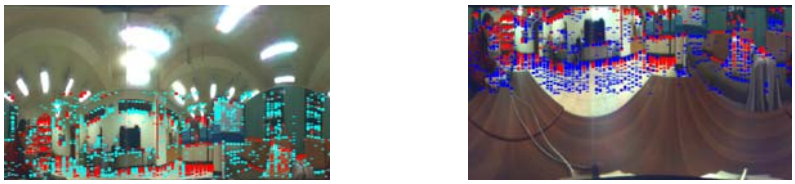


Fig. 11. Illustration of difference of elevation angles of a feature point visible in both cameras.



(a) (b)

Fig. 12. Result of dynamic programming applied to feature points in Fig. 9 with red points meaning matching points. (a) Result of upper panoramic image. (b) Result of lower panoramic image.

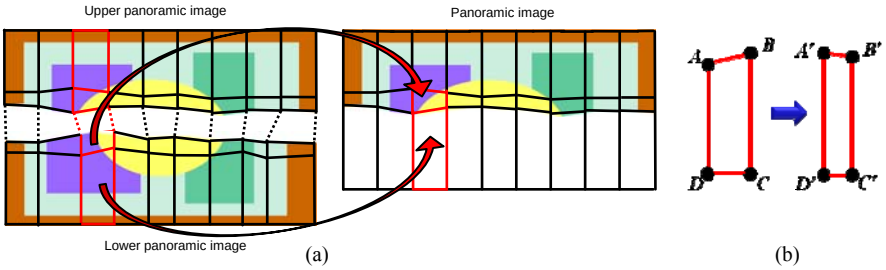
(5) Stitching of panoramic images by homographic transformation

The above steps results in a set of matching feature point pairs specified by substitutions. Then, on each matching line, we collect all the matching points and take out the two outmost ones (the highest and the lowest). Connecting these outmost points of all the matching lines in the upper and lower panoramic images, respectively, we get four piecewise-linear curves, as shown in Fig. 13(a). These curves form two quadrilaterals between every two pairs of matching lines, which we denote as Q_{up} and Q_{low}, as illustrated in Fig. 13(b). Then, we register Q_{low} and Q_{up} by homographically transforming Q_{low} to become Q_{up} [6]. Finally, we take a blending of Q_{up} and Q_{low} to yield a new quadrilateral denoted as Q_{new}. In addition, we transform similarly the patch below Q_{low}, denoted as R_{low}, to become a shape-predetermined patch, denoted as R_{up}, below Q_{up}, resulting in a new patch R_{new}. Also, let the untouched patch above Q_{up} in the upper panoramic image be denoted as S_{up}. Then, we take the final stitching result to be the union of R_{new}, Q_{new}, and S_{up}. This completes the stitching of the vertical band between a pair of matching lines. This process is repeated until all vertical bands are processed. The result of applying this stitching step to Fig. 12 is shown in Fig. 14(a). Another result is shown in Fig. 14(b) with the original omni-images and created panoramic images shown in Figs. 14(c) through 14(f).

5 Experimental results

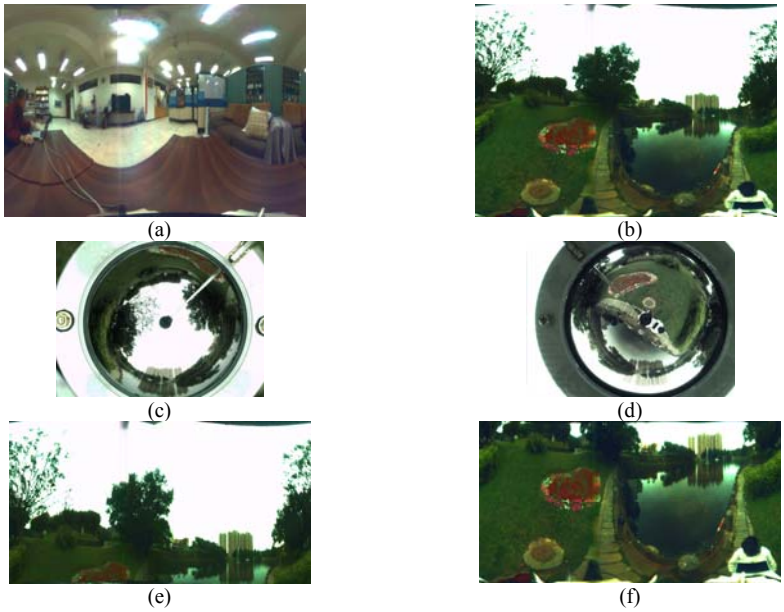
We have conducted experiments in several real environments, and some experimental results of one of them, a stairway in a library where other street view systems cannot reach, are shown here. The system interface constructed for environment navigation is shown in Fig. 15. A user first selects the environment from the database. The system then displays a perspective-view image of the first scene

spot (at the top-left corner of the interface), which is the view seen by the system operator when the environment scenes were collected during the environment learning stage. Also displayed are: (1) a walking path on the perspective-view image, which consists of step-by-step arrows representing the observable scene spots in the current view; (2) a three-view diagram showing the observable scene spots connected by line segments from three directions (the front, the top, and a side); and (3) a panoramic image of the current scene spot.



(a) (b)

Fig. 13. Stitching of panoramic images by homographic transformation. (a) Illustration of combination of stitched parts. (b) Illustration of homographic transformation between two quadrilaterals.



(a) (b)

(c) (d)

(e) (f)

Fig. 14. Results of stitching of panoramic images. (a) Result of applying homographic transformation to Fig. 12. (b) Another result of stitching with omni-images and respective panoramic images shown in (c)~(f).

Then, the user may navigate the environment further at the current spot by six ways: (1) changing the viewpoint though the perspective-view image (by clicking and dragging the image to turn into a desired view); (2) changing the viewpoint through the total panoramic image (by clicking a point on the image to select a new view); (3) going forward or backward by navigating the walking path (by clicking on an arrow of the path in Fig. 15); (4) going forward or backward step by step on the walking

path (by clicking on the red or blue arrow in Fig. 15); (5) changing the scene spot through three-view diagrams (by clicking on a point any of the three views to see the closest scene spot); (6) zooming in or out the current view (by clicking on the perspective-view image). For example, with the scene shown in Fig. 15(a) as the current spot, after the topmost arrow is clicked by Way (3) above, the next scene spot appearing in the interface is shown in Fig. 15(b).

(a) (b)

Fig. 15. Interface and environment navigation. (a) Interface for a scene spot. (b) Interface for another spot.

6 Conclusions

A new 3D imaging system with a pair of portable omni-cameras has been proposed, in which the cameras are aligned coaxially and back to back. Panoramic and perspective-view images are generated from acquired omni-image pairs based on a space-mapping approach. The overlapping band in omni-images is utilized skillfully to stitch a pair of generated panoramic images into a single one by dynamic programming for use in environment browsing. The system has several merits: (1) it is light to carry to any indoor/outdoor place; (2) it can be used to collect images in narrow, non-planar, or slanted paths where existing street view vehicles cannot navigate; (3) only two omni-images are acquired to construct various images for each scene spot; (4) perspective-view images can be generated for scene observation from any view direction; (5) six ways of environment browsing via a perspective-view image, a panoramic image, a walking path, and a three-view diagram are provided.

References

1. Wikipedia (2010, April). Google Street View. [Online]. Available: http://en.wikipedia.org/wiki/Google_Street_View.
2. Vincent, L.: Taking Online Maps Down to Street Level. *IEEE Computer*, Vol. 40, No. 12, Dec. 2007, pp. 118 – 120.
3. Jeng, S. W. and Tsai, W. H.: Using Pano-mapping Tables for Unwarping of Omni-images into Panoramic and Perspective-view Images. *IET Image Processing*, Vol. 1, No. 2, pp.149-155, June 2007.
4. Telea, A.: An Image Inpainting Technique Based on The Fast Marching Method. *Journal of Graphics Tools*, Vol. 9, No. 1, pp. 25-36, 2004.
5. He, K., Sun, J., and Tang, X.: Single Image Haze Removal Using Dark Channel Prior. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1957–1963, 2009.
6. Goes, J., et al.: *Warping & Morphing of Graphical Objects.* Morgan Kaufmann Publishers. San Francisco, USA, 1999.
7. Huaibin, Z. and Aggarwal, J. K.: 3D Reconstruction of An Urban Scene from Synthetic Fish-eye Images. *Proc. 4th IEEE Southwest Symp. on Image Analysis & Interpretation*, pp. 219-223, 2000.
8. Roman, A.: Multiperspective Imaging for Automated Urban Imaging. Doctoral dissertation, Dept. Electrical Eng., Stanford University, 2006.