Optical Engineering 48(8), 087003 (August 2009)

Unwarping of images taken by misaligned omnicameras without camera calibration by curved quadrilateral morphing using quadratic pattern classifiers

Chih-Jen Wu
National Chiao Tung University
Institute of Computer Science and Engineering
Computer Vision Laboratory
1001 University Road
Hsinchu, 30010
Taiwan

Wen-Hsiang Tsai
National Chiao Tung University
Department of Computer Science
Computer Vision Laboratory
1001 University Road
Hsinchu, 30010
Taiwan
and
Asia University
Department of Information Communication
Taichung, 41354
Taiwan
E-mail: whtsai@cis.nctu.edu.tw

Abstract. A method for solving the problem of unwarping a distorted omni-image taken by a lateral-direction misaligned omni-camera with its optical axis incoincident with its mirror axis is proposed. The method does not conduct camera calibration and is based on a new concept of two-stage image mapping from the real-world space to the distorted image space. The first stage is conducted in the camera manufacturing process and includes the generation of a pano-mapping function for mapping the real-world space to an undistorted image taken by an omni-camera with its optical and mirror axes being coincident. The second stage is conducted in an in-field environment when the omni-camera becomes lateral-direction misaligned and includes the generation of a distortion-mapping function that maps undistorted image pixels to distorted ones and the generation of a misalignment adjustment table that combines the pano-mapping and distortion-mapping functions to map the real-world space to the distorted image space. The distortion mapping function is generated by a new technique of curved quadrilateral morphing using quadratic pattern classifiers. The misalignment adjustment table is last used to unwarp distorted images conveniently by table lookup. Experimental results using simulated and real-image data show the feasibility of the proposed method. [DOI: 10.1117/1.3204238]

Subject terms: image unwarping; omni-image; omni-camera; camera calibration; camera misalignment; curved quadrilateral morphing; quadratic classifier.

Paper 081014R received Dec. 31, 2008; revised manuscript received Jun. 22, 2009; accepted for publication Jun. 24, 2009; published online Aug. 17, 2009.

1 Introduction

Omni-cameras are becoming popular for various applications owing to their advantage in providing greater fields of view (FOVs) in acquired *omni-images*. Such cameras can be categorized into two types, *dioptric* and *catadioptric*. A dioptric omni-camera captures incoming light through the camera lens to form images. An example is the fish-eye camera.[1] A catadioptric omni-camera has, in addition to a CCD camera, a reflective mirror, and captures indirect light reflected by the mirror to form images. The mirror surface may be of various shapes, such as conic, parabolic, hyperbolic, and spheric. A catadioptric omni-camera with a parabolic mirror used in this study is shown in Fig. 1, in which a transparent plastic hollow cylinder is used to support the mirror at a distance from the CCD camera placed on a platform. The structure of the camera is illustrated in Fig. 2(a). If all the reflected light rays go through a common point, the camera is said additionally to have a *single-viewpoint* (*SVP*);[2] otherwise it has a *non-single-viewpoint* (*non-SVP*).[3]

Omni-images, although providing wider FOVs, are warped in nature. In many applications, it is necessary to transform them into unwarped images. Such image unwarping work usually involves camera calibration, in which the intrinsic and extrinsic camera parameters are estimated, followed by the derivation of equations to transform image coordinates into unwarped versions.[4] The camera calibration process, presumably conducted in the camera manufacturing process, is in general complicated and time-consuming. After a calibrated camera is equipped in an application environment (e.g., installed on a vehicle, attached on a house ceiling, etc.) and used for application

Fig. 1 A catadioptric camera used in this study.

0091-3286/2009/$25.00 © 2009 SPIE

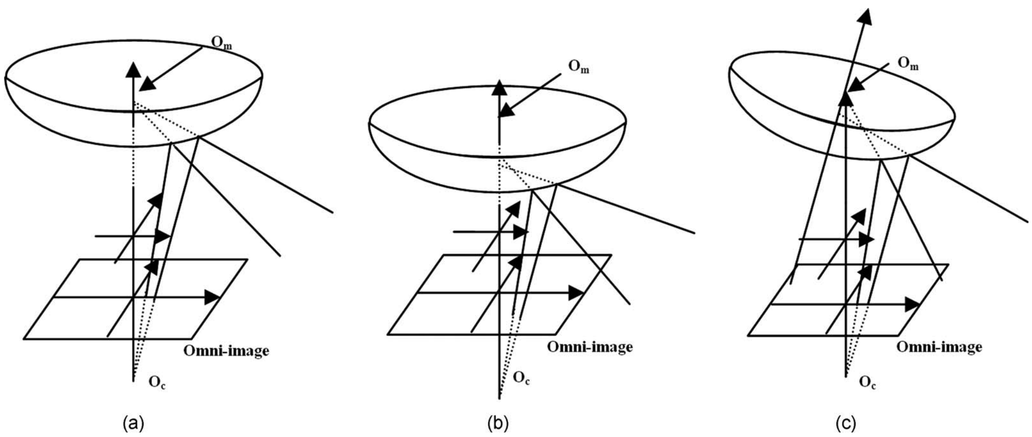


Fig. 2 Alignment of catadioptric omni-camera. (a) Correct alignment; (b) axial-direction misalignment; (c) lateral-directional misalignment.

purposes, it is usually assumed that the camera structure is fixed stably forever, incurring no change of the camera parameters.

However, in real applications like vision-based autonomous vehicle navigation or security surveillance,[1,5–8] a camera equipped on a vehicle might be shaken due to vehicle vibrations or one installed on a wall might be removed due to reemployment, causing possibly destruction of the camera structure, called *camera misalignment*, which causes displacements and/or reorientations of the CCD camera with respect to the reflective mirror. The previously mentioned non-SVP property is actually a type of camera misalignment with both the optical axis through the lens center and the mirror axis through the mirror center being axially displaced with respect to each other, resulting in destruction of the SVP into a locus called a *caustic surface*.[9] We will call such a kind of camera structure change *axial-direction camera misalignment*. An illustration is shown in Fig. 2(b). Note that usually the optical axis is assumed to be coincident with the mirror axis and that the distance of the CCD camera to the mirror surface is usually adjusted properly in advance to form the SVP property.

Another type of camera misalignment is reorientation of the CCD camera with respect to the mirror surface, resulting in destruction of the coincidence of the optical axis with the mirror axis. Such misalignment, seldom studied, destructs not only the SVP property[10] but also the rotational invariance property in omni-images, used by almost all existing image unwarping methods to simplify computation.[4,11–15] We will call such a kind of camera structure change *lateral-direction camera misalignment*. An illustration is shown in Fig. 2(c). Note that the rotational invariance property says that the angle of an incoming light ray of a scene point is identical to that of the corresponding image point in the image space. An image taken by a correctly aligned catadioptric omni-camera and another taken by a lateral-direction misaligned one are shown in Figs. 3(a) and 3(b), respectively, for illustration.

Camera misalignment causes conventional image unwarping methods inapplicable because of the resulting changes of the camera parameters. Jeng and Tsai[4] proposed an omni-image unwarping method for dealing with the axial-direction camera misalignment problem. However, very few studies on image unwarping for lateral-direction camera misalignment have been conducted so far except Mashita et al.[10] and Strelow et al.[16] In Mashita et al.,[10] camera calibration was conducted first, before image unwarping was carried out. Also, the method is applicable to cameras with hyperboloidal mirrors only. In Ref. 16, Strelow et al. proposed an imaging model for omni-cameras that accounts for the full rotation and translation between the camera and mirror if the camera is misaligned. It is required to specify manually calibration pattern points for image and world-space point correspondence.

It is desired in this study to design a general and automatic image unwarping method that can solve this problem in the application environment without camera calibration, which is usually done in the factory. Such an in-field method is useful for applications where sending misaligned cameras back to factories for recalibration is undesirable or impractical.

For this goal, the idea of a mapping-based approach proposed recently by Jeng and Tsai[11] is adopted. This approach does not conduct camera calibration to estimate camera parameters, but creates a so-called *pano-mapping table* as a substitute of camera parameters for image unwarping. It is unified and integrated in nature, applicable to unwarping of

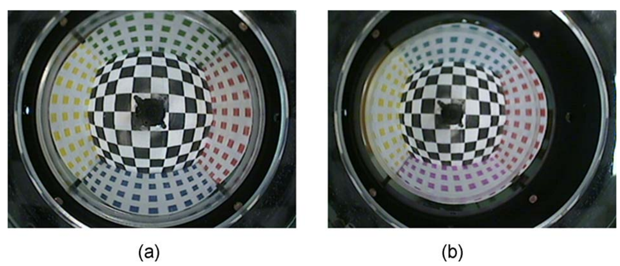


Fig. 3 Images of a color pattern acquired by a catadioptric omni-camera. (a) Image taken with the camera correctly aligned; (b) image taken with the camera misaligned.

images taken by any type of CCD camera as well as any type of reflective mirror surface.

More specifically, the proposed method has two stages, the first being assumed to be conducted in the factory and the second in the in-field environment. In the first stage, it assumed that the camera is correctly aligned to take images, which we call *undistorted images*. A pano-mapping table is then created according to Jeng and Tsai,[11] which defines a coordinate mapping function from the real-world space to the omni-image space. It can be used to unwarp an omni-image into a panoramic or a perspective-view image. In the second stage, where the camera is lateral-direction misaligned, a *distortion correction table* is created first, which maps undistorted images to distorted ones taken by the camera. The table is then combined with the pano-mapping table to create a composite mapping from the real-world space to the distorted image space, in the form of a third table, called a *misalignment adjustment table*. Such a table is used last for unwarping distorted images into panoramic images in the real-world space.

In generating the distortion correction table, which is essentially an image mapping between patches of a distorted image and those of an undistorted one, a new image morphing technique proposed in this study is applied. The technique is based on the use of the quadratic classifier in pattern recognition theory for two-class pattern classification. The use of such quadratic classifiers improves the precision of the morphing result of the conventionally adopted bilinear mapping technique, because the corresponding patches in this study have curved boundaries instead of linear ones. Furthermore, the misalignment adjustment table is invariant in nature with respective to the camera position, so the table is applicable wherever the camera is moved.

In this paper, we describe the proposed two-stage mapping-based image unwarping method as an algorithm in Sec. 2, present the proposed image patch morphing technique using quadratic classifiers in Sec. 3, show some experimental results in Sec. 4, and make concluding remarks in Sec. 5.

2 Proposed Mapping-Based Image Unwarping Method

In this section, Jeng and Tsai's method[11] used in the proposed image unwarping method is reviewed first, followed by a description of the proposed method.

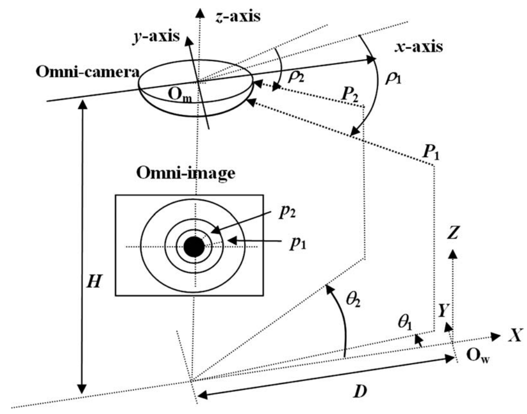


Fig. 4 Omni-camera system.

2.1 *Review of a Mapping-Based Image Unwarping Method*

The pano-mapping table proposed by Jeng and Tsai[11] is created once forever by a simple learning process for a non-lateral-direction misaligned omni-camera with any type of reflective mirror surface as a summary of the information conveyed by all the camera parameters. The learning process takes as input a set of landmark points on a calibration object in the world space and the set of corresponding points in a given image. For example, as illustrated in Fig. 4, P_1 and P_2 are two landmark points in the real world, and p_1 and p_2 are the corresponding image points, respectively. More generally, let the coordinates of each real-world point P be denoted as (θ,ρ) and those of its corresponding image pixel p as (u,v). The pair (θ,ρ) describes the azimuth angle and the elevation angle of an incident light ray coming from P and reflected by the mirror surface to go through the lens center, yielding the corresponding image pixel at (u,v) on the image plane. Accordingly, the pano-mapping table is designed to be 2-D in nature, with the horizontal and vertical axes specifying the possible ranges of θ and ρ in M and N increments, respectively, as shown in Table 1. Each entry E_{ij} with indices (i,j) in the table specifies a pair (θ_i,ρ_j), which defines an infinite set S_{ij} of real-world points on the light ray with azimuth angle θ_i and elevation angle ρ_j. These real-world points in S_{ij} are all projected onto an identical pixel p_{ij} in an omni-

Table 1 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})

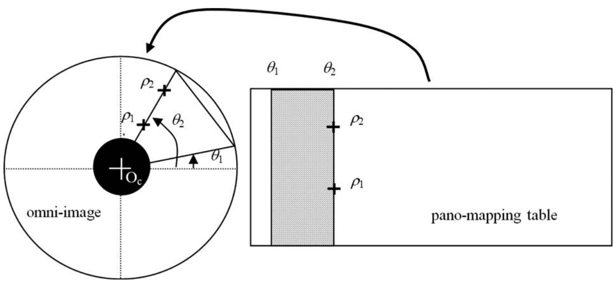


Fig. 5 Mapping between pano-mapping table and omni-image.

image taken by the camera, forming a pano-mapping, denoted as f_{pm}, from S_{ij} to p_{ij}. An illustration of this mapping is shown in Fig. 5. This mapping is shown in Table 1 by filling entry E_{ij} with the coordinates (u_{ij}, v_{ij}) of pixel p_{ij} in the omni-image.

Under the assumption of correct camera alignment that leads to the rotational invariance property, Jeng and Tsai[11] derived the following equations for computing the values (u_{ij}, v_{ij}) of each entry in the table:

$$\theta_i = i \times (2\pi/M), \quad \text{for } i = 0, 1, \ldots, M-1,$$

$$\rho_j = j \times [(\rho_e - \rho_s)/N] + \rho_s, \quad \text{for } j = 0, 1, \ldots, N-1,$$

$$r_j = f_r(\rho_j) = a_0 + a_1 \times \rho^1 + a_2 \times \rho^2 + a_3 \times \rho^3 + a_4 \rho^4,$$

$$u_{ij} = r_j \times \cos\theta_i,$$

$$v_{ij} = r_j \times \sin\theta_i, \tag{1}$$

where ρ_e and ρ_s specify the maximum and the minimum of the elevation angles of the omni-camera, respectively; $f_r(\rho)$ is a nonlinear function specifying the relation between the elevation angle ρ of a real-world point P and the radial distance r from the corresponding image pixel p at coordinates (u, v) in the omni-image to the image center; and the coefficients a_0 through a_4 of $f_r(\rho)$ are estimated using the image and real-world coordinate data of the previously mentioned corresponding landmark point pairs. Note that the rotational invariance property notationally means that the azimuth angle θ of each real-world point P on the light ray is identical to the angle ϕ of the corresponding image pixel p with respect to the u axis in the input image. That is, the azimuthal mapping is just an identity function f_a such that $f_a(\theta) = \phi = \theta$.

As illustrated laterally in Fig. 6, with the pano-mapping table T_{pm} generated as earlier and a given omni-image G, a panoramic image Q of size $M_Q \times N_Q$ with height H at distance D from the omni-camera may be generated by mapping first each image pixel q_{kl} in Q at coordinates (k, l) to an entry E_{ij} in T_{pm} filled with coordinates (u_{ij}, v_{ij}) using the parameters M_Q, N_Q, D, and H, followed by assigning the color value of the image pixel p_{ij} of G at (u_{ij}, v_{ij}) to pixel q_{kl}. The formulas for computing the indices i and j in this process are as follows:

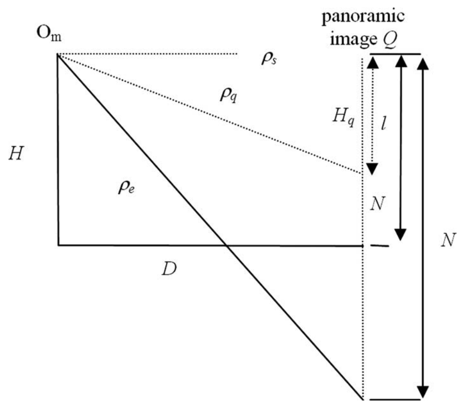


Fig. 6 Lateral-view configuration for generating a panoramic image.

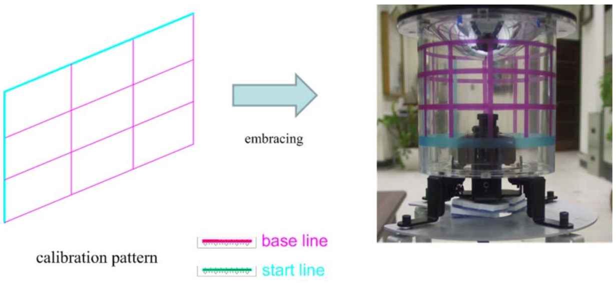
$$i = k \times \frac{M}{M_Q}; \quad H_q = l \times \frac{H}{N_Q}; \quad \rho_q = \tan^{-1}\left(\frac{H_q}{D}\right);$$
$$j = \frac{(\rho_q - \rho_s) \times N}{(\rho_e - \rho_s)}. \tag{2}$$

2.2 *Proposed Method for Unwarping Images Taken by a Misaligned Omni-Camera*

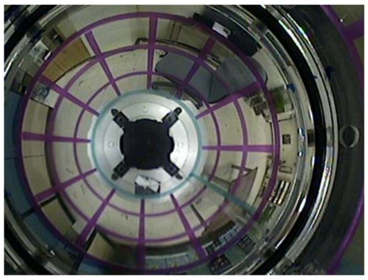
When the omni-camera is lateral-direction misaligned, as illustrated in Fig. 2(c), the image taken is distorted with respect to the undistorted image taken with a correctly aligned or axially misaligned omni-camera, as illustrated in Figs. 2(a) and 2(b), respectively. For the preceding Jeng and Tsai method,[11] which essentially is the pano-mapping f_{pm}: $(\theta_q, \rho_q) \rightarrow (u_{ij}, v_{ij})$, to be applicable, the distorted image coordinates (u'_{ij}, v'_{ij}) must be corrected in advance. From the viewpoint of image unwarping, each undistorted image pixel has a corresponding distorted one, so that there exists a distortion-mapping function f_{dm}: $(u_{ij}, v_{ij}) \rightarrow (u'_{ij}, v'_{ij})$, resulting in a composite mapping $f_{\mathrm{dm}} \circ f_{\mathrm{pm}}$: $(\theta_q, \rho_q) \rightarrow (u_{ij}, v_{ij}) \rightarrow (u'_{ij}, v'_{ij})$, or integrally, f_{ma}: $(\theta_q, \rho_q) \rightarrow (u'_{ij}, v'_{ij})$ as the overall solution to the image unwarping problem investigated in this study, where $f_{\mathrm{ma}} = f_{\mathrm{dm}} : f_{\mathrm{pm}}$. The function f_{ma} will be called the *misalignment adjustment function*.

One way to construct the nonlinear distortion-mapping function f_{dm} is to decompose the involved image part piecewise into very small patches so that the resulting subimage mappings become approximately linear. This requires implicitly the creation of a lot of feature points in the involved image part for use in the image decomposition. Instead of adopting this linearization technique, the solution proposed in this study allows the subimages to be processed nonlinearly. Such subimages come from the segmentation of the image of a calibration pattern designed for use in this study, consisting of parallel straight lines and attached on the transparent cylinder of the camera, as shown in Fig. 7(a). Each subimage is a "fan-shaped" curved quadrilateral appearing in image part of the calibration pattern, as shown in Fig. 7(b).

The calibration pattern consists of two sets of calibration



(a)



(b)

Fig. 7 Configuration of an omni-camera wrapped with a calibration pattern. (a) A calibration pattern wrapping transparent cylinder of camera; (b) image of calibration pattern consisting of "fan-shaped" curved quadrilaterals.

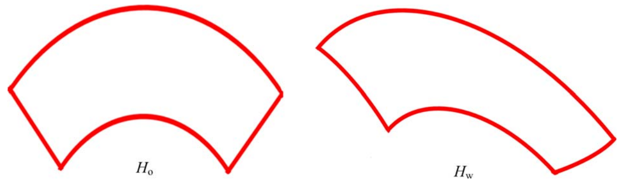


Fig. 8 Curved quadrilaterals forming a mutual corresponding pair.

lines, one set horizontal and the other set vertical. The lines are drawn in two colors: blue and purple. The blue ones, called *start lines*, are provided for facilitating line correspondence. The proposed image unwarping procedure is described in the following.

Algorithm 1: Creation of misalignment adjustment table and image unwarping. *Stage I—Generation of a pano-mapping table in the factory for a non-lateral-direction misaligned omni-camera*.

Step 1. Wrap the transparent cylinder of the camera with the calibration pattern O_c shown in Fig. 7(a), take an image of O_c as a reference image, and denote it as I_o.
Step 2. Apply Jeng and Tsai[11] to I_o to yield a pano-mapping table with mapping function f_{pm}.

Stage II—Generation of a misalignment adjustment table and unwarping of input distorted omni-images in the field.

Step 3. Wrap the transparent cylinder of the omni-camera, already lateral-direction misaligned, with the calibration pattern O_c. Take an image of O_c as a working image, and denote it as I_w.
Step 4. (Calibration line correspondence.) Perform the following steps to find corresponding calibration lines in I_o and I_w:

a. Segment the calibration lines in images I_o and I_w (appearing as curves), based on the color information (blue and purple) and the edge strengths of the lines. Classify a pixel with a sufficiently large weighted sum of its color and edge values as belonging to a calibration line.
b. Find corresponding horizontal and vertical calibration lines in I_o and I_w, respectively, by numbering the lines starting from the blue start lines. Decide two lines numbered the same from two corresponding blue start lines as a corresponding pair.

Step 5. (Curved quadrilateral correspondence.) Perform the following steps to find corresponding curved quadrilaterals in I_o and I_w, similar to those illustrated in Fig. 8:

a. Cut each corresponding horizontal calibration line pair in I_o and I_w into corresponding curve segments using the intersection points of each horizontal line with *all* the vertical calibration lines.
b. Find in order every corresponding curved quadrilateral pair in I_o and I_w by use of the corresponding curve segments.

Step 6. (Curved quadrilateral morphing.) For each pair of corresponding curved quadrilaterals, find corresponding points between them with a quadratic classification scheme (described later in the next section), resulting in a quadrilateral-morphing function.
Step 7. (Creation of a distortion-mapping function.) Perform the following steps to create a distortion-mapping function:

a. Collect all the quadrilateral-morphing functions to create a distortion-mapping function f_{dm}, which maps the coordinates of I_o to those of I_w.
b. Compose f_{dm} and f_{pm} to create a misalignment adjustment function $f_{\mathrm{ma}}=f_{\mathrm{dm}}\circ f_{\mathrm{pm}}$ in the form of a table T_{ma}, called a *misalignment adjustment table*, as the desired mapping from the real-world space to the distorted image space.

Step 8. (Unwarping input distorted omni-images.) Perform the following steps to unwarp input distorted images into panoramic ones:

a. Remove the calibration pattern O_c from the omni-camera, and take an image I_f.
b. Define a panoramic image I_p to be generated and compute the azimuth angle θ and elevation angle ρ for each pixel P in I_p according to the posture of I_p.
c. Acquire from the misalignment adjustment table T_{ma} the coordinate pair (u',v') at the entry indexed by the pair (θ,ρ).
d. Fill the pixel P in I_p with the color value of the pixel at coordinates (u',v') in I_f.

Table 2 Misalignment adjustment table of size $M \times N$.

	v_1	v_2	v_3	v_4	…	v_M
u_1	(u'_{11}, v'_{11})	(u'_{21}, v'_{21})	(u'_{31}, v'_{31})	(u'_{41}, v'_{41})	…	(u'_{M1}, v'_{M1})
u_2	(u'_{12}, v'_{12})	(u'_{22}, v'_{22})	(u'_{32}, v'_{32})	(u'_{42}, v'_{42})	…	(u'_{M2}, v'_{M2})
u_3	(u'_{13}, v'_{13})	(u'_{23}, v'_{23})	(u'_{33}, v'_{33})	(u'_{43}, v'_{43})	…	(u'_{M3}, v'_{M3})
u_4	(u'_{14}, v'_{14})	(u'_{24}, v'_{24})	(u'_{34}, v'_{34})	(u'_{44}, v'_{44})	…	(u'_{M4}, v'_{M4})
…	…	…	…	…	…	…
u_N	(u'_{1N}, v'_{1N})	(u'_{2N}, v'_{2N})	(u'_{3N}, v'_{3N})	(u'_{4N}, v'_{4N})	…	(u'_{MN}, v'_{MN})

The misalignment adjustment table created in step 7 of the preceding algorithm is designed to be of the same form as that of the pano-mapping table as shown in Table 1, except that the table entries are filled with "distorted" coordinates (u',v') of the working image and that the index for the entries is the coordinate pair (u,v) of the reference image, as in Table 2.

3 Curved Quadrilateral Morphing Using Quadratic Classifiers

In this section, the idea of the proposed technique for curved quadrilateral morphing using quadratic classifiers mentioned in step 6 of Algorithm 1 is presented first. Then the adopted quadratic classification technique[17] is reviewed, followed by a detailed description of the proposed curved quadrilateral morphing technique as an algorithm.

3.1 *Basic Idea*

A curved quadrilateral in the reference image I_o or in the working image I_w is a region enclosed by four curve boundaries. Figure 8 illustrates two corresponding curved quadrilaterals H_o and H_w from I_o and I_w, respectively. The goal of curved quadrilateral morphing is to find corresponding pixels in H_o and H_w for use in the distortion mapping function mentioned in step 7 of Algorithm 1.

Since the boundaries of the curved quadrilaterals here are all curves, the usual method of bilinear transformation for morphing quadrilaterals with line boundaries[18] is inapplicable here. It is desired to generate interpolating curves between the two opposite curves of each boundary pair, with the interpolating curves dividing each boundary curve into equal-distanced segments, as illustrated by Fig. 9. That is, for example, the curve segments a, b, c, and d of the upper boundary of the quadrilateral resulting from such boundary division in Fig. 9 are all of equal lengths, and so are the four curve segments e, f, g, and h of the left boundary of the quadrilateral. Furthermore, the curve segments formed by mutual intersections of the interpolating curves within the four boundaries also all have this equal-length property.

We propose to accomplish the preceding idea of interpolating curve generation in a recursive manner instead of directly dividing each boundary curve into a number of equal-length segments. That is, we divide recursively each of every pair of corresponding curved quadrilaterals into smaller quarter ones and consider the centers of the resulting quarter-curved quadrilaterals as corresponding points in the original corresponding curved quadrilaterals. For this purpose, we try to find the middle point of each boundary curve, resulting in two pairs of "opposite" middle boundary points. For example, in Fig. 9, the two pairs are (A,C) and (B,D). The center of a curved quadrilateral is defined to be the point at equal distances to the two middle boundary points in each pair. This point in Fig. 9 is M. It will be called the *central quadrilateral point* subsequently.

To find the middle boundary point of a curve boundary V of a curved quadrilateral, say, with two end points E and F, we adopt an approximation method as follows: (1) connect E and F into a line segment $\overline{EF}$; (2) find the bisecting point G of $\overline{EF}$; (3) find the line L_{EF} going through G and perpendicular to $\overline{EF}$, and call it the *perpendicular bisecting line* of E and F; and (4) find the intersection point S of L_{EF} and V as the desired middle boundary point. Note that S is at equal distances to A and C because of the bisection and perpendicularity property of L. The point S found in this way is just an approximation of the real middle boundary point, but it will become more accurate when the boundary curve is more symmetric with respect to the perpendicular bisecting line, as can be easily figured out. For our study here, since the distortion owing to camera misalignment is usually not too serious, this approximation is within allowable tolerance according to our experimental experience. An illustration of finding the middle boundary points of a quadrilateral is shown in Fig. 10(a).

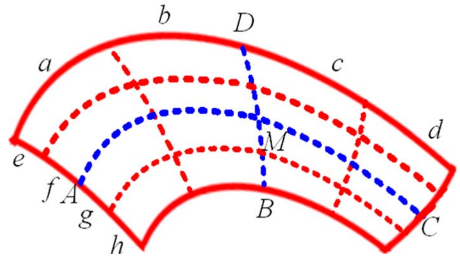


Fig. 9 Illustration of a curved quadrilateral with boundaries and interpolating curves segmented into equal-length segments (a through d are all of equal lengths; e through h are similar, and so on). (Color online only.)

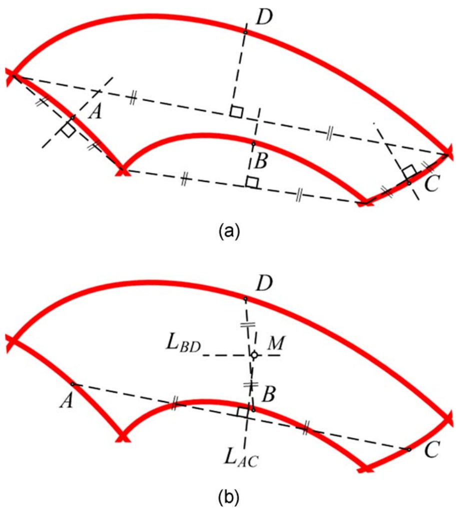


Fig. 10 Illustrations of finding the central quadrilateral point M in a curved quadrilateral. (a) Finding middle boundary points A through D by perpendicular bisecting lines of every two neighboring corners. (b) Finding central quadrilateral point M by perpendicular bisecting line of A and C, and that of B and D.

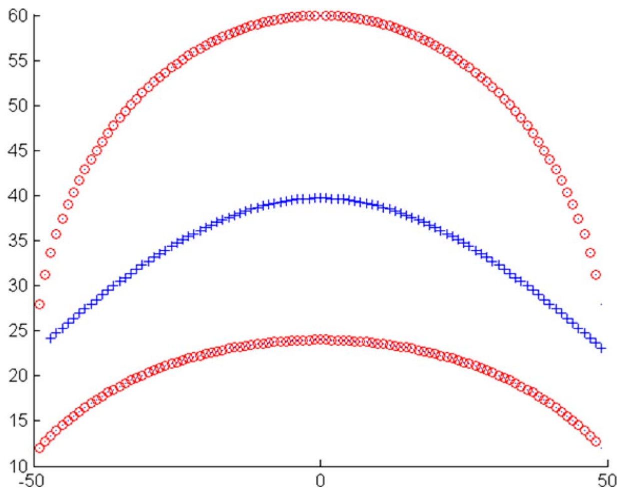


Fig. 11 Interpolating curve (blue) for two curve boundaries (red) found by a quadratic classifier using coordinate data as patterns. (The axes specify x and y coordinates.) (Color online only.)

To find the central quadrilateral point, say, for the case shown in Fig. 10(a), we adopt another approximation process that finds the perpendicular bisecting line L_{AC} of the middle boundary points A and C as well as the perpendicular bisecting line L_{BD} of the middle boundary points B and D, and then compute the intersection point M of L_{AC} and L_{BD} as the desired result. An illustration of the result of this process for Fig. 10(a) is shown in Fig. 10(b).

After the central quadrilateral point is found, the next step is to cut the original curved quadrilateral into four quarter ones. For this purpose, we have to find the curves that enclose each quarter-curved quadrilateral. Such curves should go through the central quadrilateral point, as illustrated by the two blue curves in Fig. 9. To find such interpolating curves, we adopt the quadratic classification technique used in pattern recognition theory, as reviewed in the following.

3.2 *Review of Quadratic Classification Technique*

The design of a two-class quadratic classifier in pattern recognition takes as input two sets of patterns and draws a curve as the decision boundary in the pattern space to separate the patterns into two classes in the sense of minimizing the Bayes probability of erroneously assigning the patterns into wrong classes. A good property of the classifier is its capability to generate a quadratic boundary curve that takes into consideration the shapes of the two pattern sets. That is, the decision boundary curve is roughly a blending result of the two shapes.

For our problem here, if we take each pair of "opposite" curve boundaries of a curved quadrilateral as two pattern sets with the coordinates of each boundary point as a pattern, we can design a quadratic classifier to find a decision boundary curve going through the central quadrilateral point as a desired interpolating curve, as mentioned earlier. An illustration is shown in Fig. 11, where the red coordinate data points represent two simulated boundary curves of a curved quadrilateral as input pattern sets, and the blue curve is the decision boundary of a quadratic classifier designed for the two pattern sets. Comparing the shapes of the three curves, we can see the effect of shape blending mentioned previously.

Formally, a quadratic classifier for two pattern classes ω_a and ω_b in vector form $X=[x_1 x_2]^T$ is as follows:

$$h(X) = X^T Q X + V^T X + v_o = \sum_{i=1}^{2}\sum_{j=1}^{2} q_{ij} x_i x_j + \sum_{i=1}^{2} v_i x_i + v_o, \quad (3)$$

which may be transformed into a linear form as follows:[17]

$$\begin{aligned} h(X) &= \sum_{i=i}^{3} \alpha_i y_i + \sum_{i=1}^{2} v_i x_i + v_o \\ &= [\alpha_1\ \alpha_2\ \alpha_3\ v_1\ v_2][y_1\ y_2\ y_3\ x_1\ x_2]^T + v_0 \\ &= AZ^T + v_0 \\ &= h(Z), \end{aligned}$$

where

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}, \quad V = [v_1\ v_2]^T,$$

$$A = [\alpha_1\ \alpha_2\ \alpha_3\ v_1,\ v_2]^T = [q_{11}\ q_{12} + q_{21}\ q_{22}\ v_1\ v_2]^T,$$

and

$$Z = [y_1\ y_2\ y_3\ x_1\ x_2]^T = [u^2\ uv\ v^2\ u\ v]^T.$$

By the linearity of $h(Z)=AZ^T+v_0$, we can use the design technique for the linear classifier to find the coefficient vector A and v_0. The result is as follows:

$$A = [sK_a + (1-s)K_b]^{-1}(D_b - D_a),$$

$$v_0 = -V^T[sD_a + (1-s)D_b], \tag{5}$$

where s is a scaling factor between 0 and 1 for adjusting the location of the decision boundary (normally taken to be 0.5), and D_a and D_b and K_a and K_b are the means and variances of the new pattern vectors Z^a and Z^b for classes ω_a and ω_b, respectively, which are computed as follows:

$$D_a = \frac{1}{m}\sum_{i=1}^{m} Z_i^a = \frac{1}{m}\sum_{i=1}^{m} [(u_i^a)^2 \; u_i^a v_i^a \; (v_i^a)^2 \; u_i^a \; v_i^a]^T,$$

$$D_b = \frac{1}{n}\sum_{i=1}^{n} Z_i^b = \frac{1}{n}\sum_{i=1}^{n} [(u_i^b)^2 \; u_i^b v_i^b \; (v_i^b)^2 \; u_i^b \; v_i^b]^T,$$

$$K_a = \frac{1}{m}\sum_{i=1}^{m} (Z_i^a - D_a)(Z_i^a - D_a)^T,$$

$$K_b = \frac{1}{n}\sum_{i=1}^{n} (Z_i^b - D_b)(Z_i^b - D_b)^T, \tag{6}$$

where m and n are the numbers of pattern vectors in ω_a and ω_b, respectively. By these equations, A and v_0 can be obtained and the quadratic decision boundary $h(Z)=0$, or originally $h(X)=0$, can be obtained.

3.3 *Curved Quadrilateral Morphing*

We are now ready to describe the algorithm we propose for curved quadrilateral morphing.

Algorithm 2: Quadrilateral morphing by quadratic classification.

Step 1. Acquire a curved quadrilateral H_w from the working image I_w, and its corresponding curved quadrilateral H_o from the reference image I_o.
Step 2. Compute the central quadrilateral points of H_w and H_o; denote them as M_c^w and M_c^o, respectively; and consider them as corresponding points.
Step 3. Design the quadratic classifier h_1^w for a pair of two opposite boundary curves of H_w so that the decision boundary curve of h_1^w goes through the central quadrilateral point M_c^w [i.e., $h_1^w(M_c^w)=0$] by adjusting the scaling factor s mentioned previously. Do this similarly for the other pair of opposite boundary curves of H_w to derive another quadratic classifier h_2^w which goes through M_c^w as well.
Step 4. Use the decision boundary curves of h_1^w and h_2^w to divide the curved quadrilateral H_w into four quarter ones H_1^w, H_2^w, H_3^w, and H_4^w.
Step 5. Perform steps 3 and 4 on the curved quadrilateral H_o similarly to cut H_o into four quarter-curved quadrilaterals H_1^o, H_2^o, H_3^o, and H_4^o, and take H_i^o as the curved quadrilateral corresponding to H_i^w for $i=1$, 2, 3, and 4.
Step 6. For each pair of corresponding quarter-curved quadrilaterals H_i^w and H_i^o for $i=1$, 2, 3, and 4, perform the previous steps recursively to find their respective corresponding central quadrilateral points, cut them into even smaller quarter-curved quadrilaterals, and so on, until the area of any of the quarter-curved quadrilaterals is smaller than a preselected threshold.

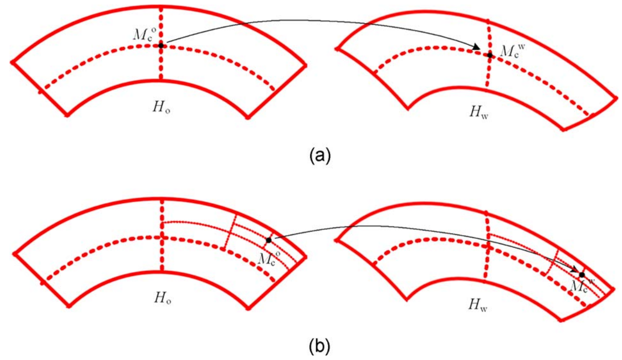


Fig. 12 Illustration of proposed algorithm for morphing one curved quadrilateral to a corresponding curved quadrilateral. (a) Finding corresponding central quadrilateral points in a pair of corresponding curved quadrilaterals. (b) Finding more corresponding central quadrilateral points within recursively cut quarter-curved quadrilaterals.

An illustration of the preceding algorithm is shown in Fig. 12. The result of the algorithm is a mapping from a set of the central quadrilateral points of subquadrilaterals of H_w to a set of the corresponding central quadrilateral points of subquadrilaterals of H_o. All the points are scattered in the original quadrilaterals H_w and H_o at nondiscrete locations. For generation of a mapping between discrete coordinates of H_w and H_o, we apply the concept of nearest neighboring to substitute nondiscrete locations with discrete ones. For example, given a point P_w in H_w with discrete coordinates (u_w, v_w), to find its corresponding point P_o in H_o with coordinates (u_o, v_o), we perform the following steps: (1) find the central quadrilateral point P_w' at nondiscrete position (u_w', v_w') in H_w that is nearest to P_w; (2) find the central quadrilateral point P_o' in H_o at nondiscrete position (u_o', v_o') corresponding to P_w'; and (3) find the point P_o at discrete position (u_o, v_o) in H_o that is nearest to P_o'. That is, we have the following series of mappings:

$$P_w(u_w, v_w) \to P_w'(u_w', v_w') \to P_o'(u_o', v_o') \to P_o(u_o, v_o), \tag{7}$$

and we take the overall mapping $(u_w, v_w) \to (u_o, v_o)$ as the final result.

4 Experimental Results

A series of experiments has been conducted to verify the proposed method. The first experiment was conducted to test the proposed morphing algorithm (Algorithm 2) on some simulated data of curved quadrilaterals, which were drawn by hand and transformed into images. Figure 13 shows one of the intermediate results of iterative central quadrilateral point computation, where a given input curved quadrilateral is shown in Fig. 13(a), and the results of the first three iterations are shown in Figs. 13(b)–13(d), respectively. In each figure, the dark blue lines are the interpolating curves found by quadratic classifications, and the light blue points are the found central quadrilateral points.

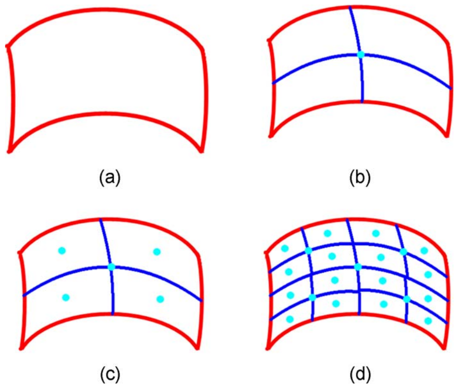


Fig. 13 Results of curved quadrilateral morphing by Algorithm 2 using simulated data. (a) A simulated curved quadrilateral. (b) Result of first iteration. (c) Result of second iteration. (d) Result of third iteration. (Color online only.)

In the second experiment, we use real data (a pair of curved quadrilaterals taken from real omni-images acquired by a hyperbolic catadioptric omni-camera manufactured by Micro-Star International Co.) to conduct the same process of the first experiment described earlier. The results are shown in Fig. 14, with Figs. 14(a)–14(d) corresponding respectively to Figs. 13(a)–13(d) in meaning.

In the third experiment, we tested Algorithm 1 and Algorithm 2 together using real-image data. The calibration pattern we used is the one attached on the transparent cylinder of the omni-camera shown in Fig. 7(a). An undistorted image of the calibration pattern assumed to be taken in a factory for use as the reference image is shown in Fig. 15(a). And a distorted version of Fig. 15(a) taken in field using a lateral-direction misaligned omni-camera for use as the working image is shown in Fig. 15(b). Figures 15(c) and 15(d) are intermediate results that show the segmented calibration lines of Figs. 15(a) and 15(b), respectively. Using the two figures and subsequent results, a misalignment adjustment table was created by stage II of Algorithm 1. And Fig. 15(e) is the result of applying Algorithm 2 to Fig. 15(b) using this table. This figure was used further to generate a panoramic image, which is shown in Fig. 15(f).

As a contrast, we also generated a panoramic image from the distorted image of Fig. 15(b) without using the distortion mapping f_{dm} (i.e., using the pano-mapping f_{pm} only), and the result is shown in Fig. 15(g), which is quite unacceptable. This means that the proposed approach is quite significant in correcting image distortion caused by lateral-direction camera misalignment.

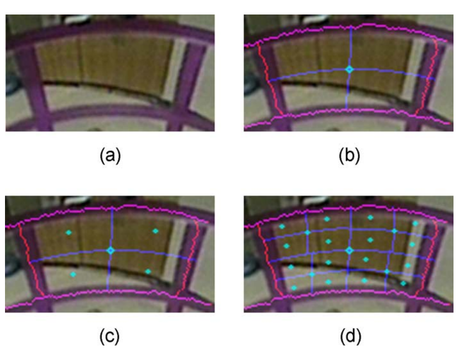


Fig. 14 Results of quadrilateral morphing for real data using Algorithm 2. (a) A curved quadrilaterals. (b) Result of first iteration. (c) Result of second iteration. (d) Result of third iteration with intermediate curves removed.

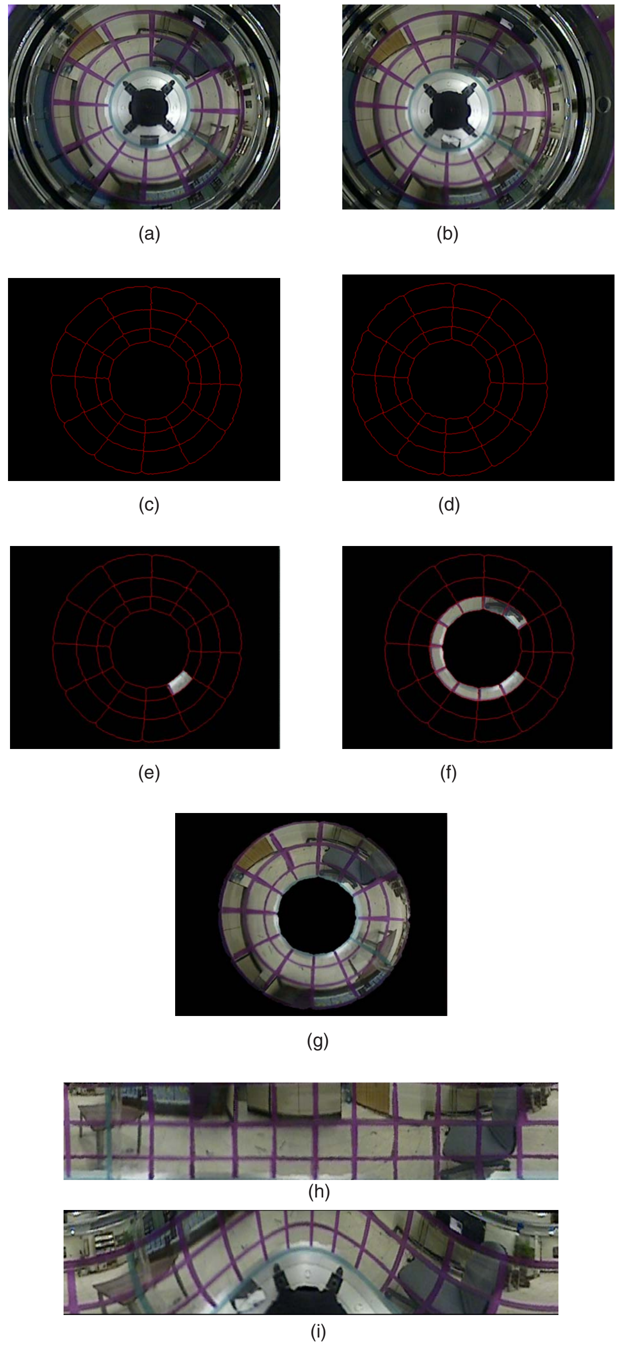


Fig. 15 Image unwarping results using Algorithm 1 and Algorithm 2. (a) Reference image. (b) Working image. (c) Segmented calibration lines in (a) in thinned form. (d) Segmented calibration lines in (b) in thinned form. (e) One patch of intermediate results of applying Algorithm 2 to (b) using the misalignment adjustment table. (f) Ten patches of intermediate results of applying Algorithm 2 to (b) using the misalignment adjustment table. (g) Result of applying Algorithm 2 to (b) using the misalignment adjustment table. (h) A panoramic image generated from (g). (i) A panoramic image generated from (b).

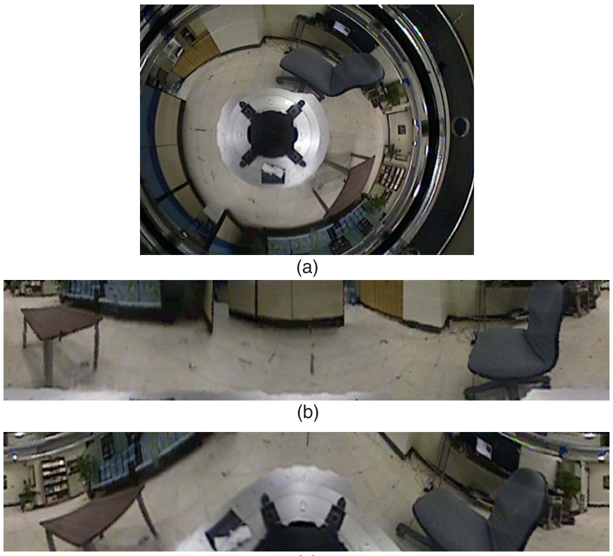

(a)
(b)
(c)

Fig. 16 Results of distorted image unwarping using Algorithm 2. (a) Distorted omni-image. (b) Created panoramic image with correction by Algorithm 2. (c) Created panoramic image without misalignment correction.

In the fourth experiment, we tested the effect of Algorithm 2 on the real-image data misalignment adjustment table obtained in the previous experiment. The result is shown in Fig. 16. The input distorted image, shown in Fig. 16(a), was acquired from the lateral-direction misaligned camera used in the third experiment with the calibration pattern removed. The panoramic image created from this distorted omni-image using Algorithms 1 and 2 is shown in Fig. 16(b). For comparison again, we create another panoramic image using the pano-mapping function only without distortion correction, and the result is shown in Fig. 16(c). Comparing Fig. 16(b) with Fig. 16(c), we see again that the proposed method is effective.

5 Conclusions

A method for solving the new problem of unwarping omni-images taken by lateral-direction misaligned omni-cameras is proposed, which is based on the concept of direct pixel mapping instead of the conventional camera calibration approach. The method may be regarded as a generalization of that proposed by Jeng and Tsai[11] which is applicable to images taken by omni-cameras with no lateral-direction misalignment. The proposed method uses a misalignment adjustment table to map real-world space points to distorted image pixels in a table lookup manner, thus speeding up the image unwarping process. The table is a composite of a distortion-mapping function defined in this study and a pano-mapping function generated by the Jeng and Tsai method. The distortion-mapping function is generated by a new technique of curved quadrilateral morphing based on quadratic classification in pattern recognition theory. Quadratic classifiers are used to generate interpolating curves within corresponding curved quadrilaterals extracted from the reference and working images. Such curves are used to generate corresponding pixels in distorted and undistorted images, which are then used for generating the content of the misalignment adjustment table. Experimental results show the feasibility of the proposed method.

In Ref. 19, Ainouz et al. proposed a novel idea for warping pixel-neighborhood-based operators for image derivation, image convolution, image matching, etc. to adapt them to distorted omni-images. A possible future study may be directed to applying the curved quadrilateral morphing using quadratic pattern classifiers proposed in our method for the same purpose of warping the operators. Other possible future studies include investigating the possibility of deriving the misalignment adjustment table directly without combining two tables and using the proposed curved quadrilateral morphing technique for unwarping other types of image distortion.

Acknowledgments

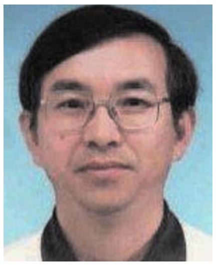
This work was supported financially by the Ministry of Economic Affairs under Project No. MOEA 97-EC-17-A-02-S1-032 in the Technology Development Program for Academia.

References

1. Y. C. Liu, K. Y. Lin, and Y. S. Chen, "Bird's-eye view vision system for vehicle surrounding monitoring," in *Proc. Conf. Robot Vision*, *Lect. Notes Comput. Sci.* **4931**, pp. 207–218 (2008).
2. S. Baker and S. K. Nayar, "A theory of single-viewpoint catadioptric image formation," *Int. J. Comput. Vis.* **35**(2), 175–196 (1999).
3. J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," *IEEE Trans. Rob. Autom.* **16**(6), 890–898 (2000).
4. S. W. Jeng and W. H. Tsai, "Analytic image unwarping by a systematic calibration method for omni-directional cameras with hyperbolic-shaped mirrors," *Image Vis. Comput.* **26**(5), 690–701 (2008).
5. S. Hrabar and G. S. Sukhatme, "Omnidirectional vision for an autonomous helicopter," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3602–3609, IEEE, Piscataway, NJ (2003).
6. G. Tarak and M. M. Trivedi, "Parametric ego-motion estimation for vehicle surround analysis using omni-directional camera," *Mach. Vision Appl.* **16**(2), 85–95 (2005).
7. M. Senoh, F. Kozawa, and M. Yamada, "Development of shape measurement system using an omnidirectional sensor and light sectioning method with laser beam scanning for Hume pipes," *Opt. Eng.* **45**(6), 064301 (2006).
8. T. Boult, "DOVE: dolphin omni-directional video equipment," in *Proc. 8th IASTED Int. Conf. Robotics and Automation*, pp. 214–220, Acta Press, Calgary, Canada (2000).
9. R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "Caustics of catadioptric cameras," in *Proc. Eighth IEEE International Conf. Computer Vision*, vol. **2**, pp. 2–9, IEEE, Piscataway, NJ (2001).
10. T. Mashita, Y. Iwai, and M. Yachida, "Calibration method for misaligned catadioptric camera," *IEICE Trans. Inf. Syst.* **E89-D**(7), 1984–1993 (2006).
11. S. W. Jeng and W. H. Tsai, "Using pano-mapping tables for unwarping omni-images into panoramic and perspective-view images," *IET Image Process.* **1**(2), 149–155 (2007).
12. S. K. Nayar, "Catadioptric omni-directional camera," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 482–488 (1997).
13. Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, "Visual surveillance and monitoring system using an omni-directional video camera," in *Proc. 14th Int. Conf. Pattern Recognition*, vol. **1**, pp. 588–592 (1998).
14. X. H. Ying and Z. Y. Hu, "Catadioptric camera calibration using geometric invariants," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(10), 1260–1271 (2004).
15. S. W. Jeng and W. H. Tsai, "Construction of perspective and panoramic images from omni-images taken from hypercatadioptric cameras for visual surveillance," in *Proc. 2004 IEEE Int. Conf. Networking, Sensing, and Control*, pp. 204–209 (2004).
16. D. Strelow, J. Mishler, D. Koes, and S. Singh, "Precise omnidirectional camera calibration," in *Proc. 2001 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR 2001)*, pp. 689–694 (2001).
17. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, New York (1990).

18. J. Gomes, L. Darsa, B. Costa, and L. Velho, *Warping and Morphing of Graphical Objects*, Morgan Kauffmann Publishers, San Francisco (1999).
19. S. Ainouz, O. Morel, and D. Fofi, "Mirror-adapted matching of catadioptric images," in *IEEE Int. Conf. Image Processing (ICIP'08)* (2008).

Chih-Jen Wu received his BS degree and MS degree, both in engineering science, from National Cheng Kung University, Tainan, Taiwan, in 2000 and 2002, respectively. Mr. Wu worked as a research assistant in the Laboratory of System Integration at National Cheng Kung University from 2000 to 2002 and as a research engineer in the Computer Vision Laboratory at National Chiao Tung University from 2002 to now. He has been a PhD student in the Department of Computer Science at National Chiao Tung University since 2002. His current research interests include computer vision, robotics, pattern recognition, and their applications.



Wen-Hsiang Tsai received a BS degree in electrical engineering from National Taiwan University in 1973, an MS degree in electrical engineering from Brown University in 1977, and a PhD degree in electrical engineering from Purdue University in 1979. Dr. Tsai joined the faculty of National Chiao Tung University (NCTU) in Taiwan in 1979 and is now an NCTU Chair Professor in the Department of Computer and Information Science. At NCTU, Professor Tsai was the head of the Department of Computer and Information Science from 1984 to 1988, the dean of general affairs from 1995 to 1996, the dean of academic affairs from 1999 to 2001, and vice president from 2001 to 2004. From 2004 to 2007, he was president of Asia University in Taiwan. He served as the chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan from 1999 to 2000 as well as the chair of the Computer Society of IEEE Taipei Section in Taiwan from 2003 to 2008. He has been the editor of several academic journals, including *Journal of Chinese Engineers, International Journal of Pattern Recognition and Artificial Intelligence, Journal of Information Science and Engineering*, and *Pattern Recognition*. He was editor-in-chief of *Journal of Information Science and Engineering* from 1998 to 2000 and of *Asian Journal of Health and Information Sciences* from 2006 to 2009. Professor Tsai has received numerous awards, including awards from the National Science Council, the Pattern Recognition Society, and the Ministry of Education. Professor Tsai's major research interests include image processing, pattern recognition, computer vision, virtual reality, and information copyright and security protection. He has published 344 academic papers, including 132 journal papers.