

Flexible Camera Calibration Using a New Analytical Radial Undistortion Formula with Application to Mobile Robot Localization

Lili Ma, *Student Member, IEEE*, YangQuan Chen and Kevin L. Moore, *Senior Members, IEEE*
Center for Self-Organizing and Intelligent Systems (CSOIS),
Dept. of Electrical and Computer Engineering, 4160 Old Main Hill,
Utah State University (USU), Logan, UT 84322-4160, USA.
Emails: lilima@cc.usu.edu, {yqchen, moorek}@ece.usu.edu

Abstract—Most algorithms in 3D computer vision rely on the pinhole camera model because of its simplicity, whereas virtually all imaging devices introduce certain amount of nonlinear distortion, where the radial distortion is the most severe part. Common approach to radial distortion is by the means of polynomial approximation, which introduces distortion-specific parameters into the camera model and requires estimation of these distortion parameters. The task of estimating radial distortion is to find a radial distortion model that allows easy undistortion as well as satisfactory accuracy. This paper presents a new radial distortion model with an easy analytical undistortion formula, which also belongs to the polynomial approximation category. Experimental results are presented to show that with this radial distortion model, satisfactory accuracy is achieved. An application of the new radial distortion model is non-iterative yellow line alignment with a calibrated camera on ODIS, a robot built in our CSOIS (See Fig. 1).

I. INTRODUCTION

A. Related Work: Camera Calibration

Depending on what kind of calibration object used, there are mainly two categories of calibration methods: photogrammetric calibration and self-calibration. Photogrammetric calibration refers to those methods that observe a calibration object whose geometry in 3-D space is known with a very good precision [1]. Self-calibration does not need any calibration object. It only requires point matches from image sequence. In [2], it is shown that it is possible to calibrate a camera just by pointing it to the environment, selecting points of interest and then tracking them in the image as the camera moves. The obvious advantage of the self-calibration method is that it is not necessary to know the camera motion and it is easy to set up. The disadvantage is that it is usually considered unreliable [3]. A four step calibration procedure is proposed in [4] where the calibration is performed with a known 3D target. The four steps in [4] are: linear parameter estimation, nonlinear optimization, correction using circle/ellipse, and image correction. But for a simple start, linear parameter estimation and nonlinear optimization are enough. In [5], a plane-based calibration method is described where the calibration is performed by first determining the absolute conic $B = A^{-T}A^{-1}$, where A is a matrix formed by the camera's intrinsic parameters. In [5], the parameter γ (a

parameter describing the skewness of the two image axes) is assumed to be zero and it is observed that only the relative orientations of planes and camera are of importance in avoiding singularities because the planes that are parallel to each other provide exactly the same information. The camera calibration method in [6], [7] is regarded as a great contribution to the camera calibration. It focuses on the desktop vision system and advances 3D computer vision one step from laboratory environments to the real world. The proposed method in [6], [7] lies between the photogrammetric calibration and the self-calibration, because 2D metric information is used rather than 3D. The key feature of the calibration method in [6], [7] is that the absolute conic B is used to estimate the intrinsic parameters and the parameter γ can be considered. The proposed technique in [6], [7] only requires the camera to observe a planar pattern at a few (at least 3, if both the intrinsic and the extrinsic parameters are to be estimated uniquely) different orientations. Either the camera or the calibration object can be moved by hand as long as they cause no singularity problem and the motion of the calibration object or camera itself needs not to be known in advance.

After estimation of camera parameters, a projection matrix M can directly link a point in the 3-D world reference frame to its projection (undistorted) in the image plane. That is

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = A [R \ t] \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R \ t] \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix}, \quad (1)$$

where λ is an arbitrary scaling factor and the matrix A fully depends on the 5 intrinsic parameters with their detail descriptions in Table I, where some other variables used throughout this paper are also listed.

The calibration method used in this work is to first estimate the projection matrix and then use the absolute conic to estimate the intrinsic parameters [6], [7]. The detail procedures are summarized below:

- Linear Parameter Estimation,
 - Estimation of Intrinsic Parameters;
 - Estimation of Extrinsic Parameters;
 - Estimation of Distortion Coefficients;
- Nonlinear Optimization.

TABLE I
LIST OF VARIABLES

Variable	Description
$P^w = [X^w, Y^w, Z^w]^T$	3-D point in world frame
$P^c = [X^c, Y^c, Z^c]^T$	3-D point in camera frame
$k = (k_1, k_2)$	Distortion coefficients
(u_d, v_d)	Distorted image points
(u, v)	Undistorted image points
(x_d, y_d)	$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix}$
(x, y)	$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$
r	$r^2 = x^2 + y^2$
$\alpha, \beta, \gamma, u_0, v_0$	5 intrinsic parameters
J	Objective function
$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$	Camera intrinsic matrix
$B = A^{-T} A^{-1}$	Absolute conic
M	Projection matrix

B. Radial Distortion

Radial distortion causes an inward or outward displacement of a given image point from its ideal location. The negative radial displacement of the image points is referred to as the barrel distortion, while the positive radial displacement is referred to as the pincushion distortion [8]. The radial distortion is governed by the following equation [6], [8]:

$$F(r) = r f(r) = r(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots), \quad (2)$$

where k_1, k_2, k_3, \dots are the distortion coefficients and $r^2 = x^2 + y^2$ with (x, y) the normalized undistorted projected points in the camera frame. The distortion is usually dominated by the radial components, and especially dominated by the first term. It has also been found that too high an order in (2) may cause numerical instability [7], [9], [10]. In this paper, at most two terms of radial distortion are considered. When using two coefficients, the relationship between the distorted and the undistorted image points becomes [6]

$$\begin{aligned} u_d - u_0 &= (u - u_0)(1 + k_1 r^2 + k_2 r^4) \\ v_d - v_0 &= (v - v_0)(1 + k_1 r^2 + k_2 r^4). \end{aligned} \quad (3)$$

When using two distortion coefficients to model radial distortion as in [6], [11], the inverse of the polynomial function in (3) is difficult to perform analytically. In [11], the inverse

function is obtained numerically via an iterative scheme. In [12], for practical purpose, only one distortion coefficient k_1 is used. Besides the polynomial approximation method mentioned above, a technique for blindly removing lens distortion in the absence of any calibration information in the frequency domain is presented in [13]. However, the accuracy reported in [13] is by no means comparable to that based on calibration and this approach can be useful in areas where only qualitative results are required. The new radial distortion model proposed in this paper belongs to the polynomial approximation category.

The rest of the paper is organized as follows. Sec. II describes the new radial distortion model and its inverse undistortion analytical formula. Experimental results and comparison with existing models are presented in Sec. III. One direct application of this new distortion model is discussed in Sec. IV. Finally, some concluding remarks are given in Sec. V.

II. RADIAL DISTORTION MODELS

In this paper, we focus on the distortion models while the intrinsic parameters and the extrinsic parameters are achieved using the method presented in [6], [7]. According to the radial distortion model in (3), the radial distortion can be resulted in one of the following two ways:

- Transform from the camera frame to the image plane, then perform distortion in the image plane

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} u_d \\ v_d \end{bmatrix};$$

- Perform distortion in the camera frame, then transform to the image plane

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x_d \\ y_d \end{bmatrix} \rightarrow \begin{bmatrix} u_d \\ v_d \end{bmatrix},$$

where

$$x_d = x f(r), \quad y_d = y f(r). \quad (4)$$

Since

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

(3) becomes

$$\begin{aligned} u_d &= (u - u_0) f(r) + u_0 \\ &= \alpha x f(r) + \gamma y f(r) + u_0 \\ &= \alpha x_d + \gamma y_d + u_0, \\ v_d &= (v - v_0) f(r) + v_0 \\ &= \beta y_d + v_0. \end{aligned} \quad (5)$$

Therefore, it is also true that

$$\begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = A \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}.$$

Thus, the distortion performed in the image plane can also be understood as introducing distortion in the camera frame and then transform back to the image plane.

A. The Existing Radial Distortion Models

Radial undistortion is to extract (u, v) from (u_d, v_d) , which can also be accomplished by extracting (x, y) from (x_d, y_d) . The following derivation shows the problem when trying to extract (x, y) from (x_d, y_d) using two distortion coefficients k_1 and k_2 in (3).

From (u_d, v_d) , we can calculate (x_d, y_d) by

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & -\frac{\gamma}{\alpha\beta} & -\frac{u_0}{\alpha} + \frac{v_0\gamma}{\alpha\beta} \\ 0 & \frac{1}{\beta} & -\frac{v_0}{\beta} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix}, \quad (6)$$

where the camera intrinsic matrix \mathbf{A} is invertible by nature. Now, the problem becomes to extracting (x, y) from (x_d, y_d) . According to (4),

$$\begin{aligned} x_d &= x f(r) = x[1 + k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ y_d &= y f(r) = y[1 + k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]. \end{aligned} \quad (7)$$

It is obvious that $x_d = 0$ iff $x = 0$. When $x_d \neq 0$, by letting $c = y_d/x_d = y/x$, we have $y = cx$ where c is a constant. Substituting $y = cx$ into the above equation gives

$$\begin{aligned} x_d &= x[1 + k_1(x^2 + c^2x^2) + k_2(x^2 + c^2x^2)^2] \\ &= x + k_1(1 + c^2)x^3 + k_2(1 + c^2)^2x^5. \end{aligned} \quad (8)$$

Let $f(x) = x + k_1(1 + c^2)x^3 + k_2(1 + c^2)^2x^5$. Then $f(-x) = -f(x)$ and $f(x)$ is an odd function. The analytical solution of (8) is not a trivial task. This analytical problem is still open (of course, we can use numerical method to solve it). But if we set $k_2 = 0$, the analytical solution is available and the radial undistortion can be done easily. In [12], for the same practical reason, only one distortion coefficient k_1 is used to approximate the radial distortion, in which case we would expect to see performance degradation. In Sec. III, experimental results are presented to show the performance comparison for the cases when $k_2 = 0$ and $k_2 \neq 0$ using the calibrated parameters of three different cameras. Recall that the initial guess for radial distortion is done after having estimated all other parameters (including both intrinsic and extrinsic parameters) and just before the nonlinear optimization step. So, we can reuse the estimated parameters and choose the initial guess for k_2 to be 0 and compare the values of objective function after nonlinear optimization.

The objective function used for nonlinear optimization is [6]:

$$J = \sum_{i=1}^N \sum_{j=1}^n \|m_{ij} - \hat{m}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j)\|^2, \quad (9)$$

where $\hat{m}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j)$ is the projection of point M_j in the i^{th} image using the estimated parameters and M_j is the j^{th} 3D point in the world frame with $Z^w = 0$. Here, n is the number of feature points in the coplanar calibration object and N is the number of images taken for calibration.

B. The New Radial Distortion Model

Our new radial distortion model is proposed as:

$$F(r) = r f(r) = r(1 + k_1r + k_2r^2), \quad (10)$$

which is also a function only related to radius r . The motivation of choosing this radial distortion model is that the resultant approximation of x_d is also an odd function of x , as can be seen next. For $F(r) = r f(r) = r(1 + k_1r + k_2r^2)$, we have

$$\begin{aligned} x_d &= x f(r) = x(1 + k_1r + k_2r^2) \\ y_d &= y f(r) = y(1 + k_1r + k_2r^2). \end{aligned} \quad (11)$$

Again, let $c = y_d/x_d = y/x$. We have $y = cx$ where c is a constant. Substituting $y = cx$ into the above equation gives

$$\begin{aligned} x_d &= x \left[1 + k_1\sqrt{x^2 + c^2x^2} + k_2(x^2 + c^2x^2) \right] \\ &= x \left[1 + k_1\sqrt{1 + c^2} \operatorname{sgn}(x)x + k_2(1 + c^2)x^2 \right] \\ &= x + k_1\sqrt{1 + c^2} \operatorname{sgn}(x)x^2 + k_2(1 + c^2)x^3, \end{aligned} \quad (12)$$

where $\operatorname{sgn}(x)$ gives the sign of x . Let

$$f(x) = x + k_1\sqrt{1 + c^2} \operatorname{sgn}(x)x^2 + k_2(1 + c^2)x^3.$$

Clearly, $f(x)$ is also an odd function.

To perform the radial undistortion using the new distortion model in (10), that is to extract x from x_d in (12), the following algorithm is applied:

- 1) $x = 0$ iff $x_d = 0$,
- 2) Assuming that $x > 0$, (12) becomes

$$x_d = x + k_1\sqrt{1 + c^2}x^2 + k_2(1 + c^2)x^3.$$

Using solve, a Matlab Symbolic Toolbox function, we can get three possible solutions for the above equation denoted by x_{1+} , x_{2+} , and x_{3+} respectively. To make the equations simple, let $y = x_d$, $p = k_1\sqrt{1 + c^2}$ and $q = k_2(1 + c^2)$. The three possible solutions for $y = x + px^2 + qx^3$ are

$$\begin{aligned} x_{1+} &= \frac{1}{6q}E_1 + \frac{2}{3}E_2 - \frac{p}{3q}, \\ x_{2+} &= -\frac{1}{12q}E_1 - \frac{1}{3}E_2 - \frac{p}{3q} + \frac{\sqrt{3}}{2}\left(\frac{1}{6q}E_1 - \frac{2}{3}E_2\right)j, \\ x_{3+} &= -\frac{1}{12q}E_1 - \frac{1}{3}E_2 - \frac{p}{3q} - \frac{\sqrt{3}}{2}\left(\frac{1}{6q}E_1 - \frac{2}{3}E_2\right)j, \end{aligned} \quad (13)$$

where

$$\begin{aligned} E_1 &= \{36pq + 108yq^2 - 8p^3 \\ &\quad + 12\sqrt{3}q\sqrt{4q - p^2 + 18pqy + 27y^2q^2 - 4yp^3}\}^{1/3}, \\ E_2 &= \frac{p^2 - 3q}{qE_1}, \quad j = \sqrt{-1}. \end{aligned} \quad (14)$$

From the above three possible solutions, we discard those whose imaginary parts are not equal to zero. Then, from the remaining, discard those solutions that conflict with the assumption that $x > 0$. Finally, we get the candidate

solution x_+ by choosing the one closest to x_d if the number of remaining solutions is greater than 1.

3) Assuming that $x < 0$, there are also three possible solutions for

$$x_d = x - k_1 \sqrt{1 + c^2} x^2 + k_2(1 + c^2) x^3, \quad (15)$$

which can be written as

$$y = x + (-p)x^2 + qx^3. \quad (16)$$

The three solutions for (16) can thus be calculated from (13) and (14) by substituting $p = -p$. With a similar procedure as described in the case for $x > 0$, we will have another candidate solution x_- .

4) Choose among x_+ and x_- for the final solution of x by taking the one closest to x_d .

The basic idea to extract x from x_d in (12) is to choose from several candidate solutions, whose analytical formula are known. The benefits of using this new radial distortion model are as follows:

- Low order fitting, better for fixed-point implementation;
- Explicit or analytical inverse function with no numerical iterations;
- Better accuracy than using the radial distortion model $f(r) = 1 + k_1 r^2$.

III. EXPERIMENTAL RESULTS AND COMPARISONS

Now, we want to compare the performance of three different radial distortion models based on the final value of objective function after nonlinear optimization by the Matlab function `fminunc`. The three different distortion models for comparison are:

$$\text{distortionmodel}_1: f(r) = 1 + k_1 r^2 + k_2 r^4,$$

$$\text{distortionmodel}_2: f(r) = 1 + k_1 r^2,$$

$$\text{distortionmodel}_3: f(r) = 1 + k_1 r + k_2 r^2.$$

Using the public domain test images [14], the desktop camera images [15] (a color camera in our CSOIS), and the ODIS camera images [15] (the camera on ODIS robot built in our CSOIS, see Sec. IV-A and Fig. 1), the final objective function (J), the 5 estimated intrinsic parameters ($\alpha, \beta, \gamma, u_0, v_0$), and the estimated distortion coefficients (k_1, k_2) are shown in Tables II, III, and IV respectively [15]. The results show that the objective function of model_3 is always greater than that of model_1 , but much smaller than that of model_2 , which is consistent with our expectation. Note that, when doing nonlinear optimization with different distortion models, we always use the same exit thresholds.

To make the results in this paper repeatable by other researchers for further investigation, we present the options we use for the nonlinear optimization: `options = optimset('Display', 'iter', 'LargeScale', 'off', 'MaxFunEvals', 8000, 'TolX', 10^{-5} , 'TolFun', 10^{-5} , 'MaxIter', 120)`. The raw data of the extracted feature locations in the image plane are also available upon request.

A second look at the results reveals that for the camera used in [6], [7], [14], which has a small lens distortion, the advantage of model_3 over model_2 is not so significant. When the cameras are experiencing severe distortion, the radial distortion model_3 gives a much better performance over model_2 , as can be seen from Tables III and IV.

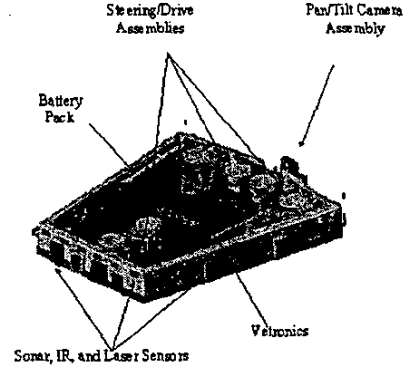


Fig. 1. The mechanical and vetronics layout of ODIS

TABLE II
COMPARISON OF DISTORTION MODELS USING IMAGES IN [14] *

Microsoft Images			
Model	#1	#2	#3
J	144.88	148.279	145.659
α	832.5010	830.7340	833.6623
γ	0.2046	0.2167	0.2074
u_0	303.9584	303.9583	303.9771
β	832.5309	830.7898	833.6982
v_0	206.5879	206.5692	206.5520
k_1	-0.2286	-0.1984	-0.0215
k_2	0.1903	0	-0.1565

* (k_1, k_2) for model_1 and model_2 are defined in (2) and (k_1, k_2) for model_3 is defined in (10).

IV. APPLICATION: NON-ITERATIVE YELLOW LINE ALIGNMENT WITH A CALIBRATED CAMERA ON ODIS

A. What is ODIS?

The Utah State University Omni-Directional Inspection System (USU ODIS) is a small, man-portable mobile robotic system that can be used for autonomous or semi-autonomous inspection under vehicles in a parking area [16], [17], [18]. The robot features (a) three "smart wheels" [19] in which both the speed and direction of the wheel can be independently controlled through dedicated processors, (b) a vehicle electronic capability that includes multiple processors, and (c) a sensor array with a laser, sonar and IR sensors, and a video camera. A unique feature in ODIS is the notion of the "smart wheel" developed by the Center for Self-Organizing and Intelligent Systems

TABLE III
Comparison of Distortion Models Using Desktop Images in [15]

Desktop Images			
Model	#1	#2	#3
J	778.9768	904.68	803.307
α	277.1457	275.5959	282.5664
γ	-0.5730	-0.6665	-0.6201
u_0	153.9923	158.2014	154.4891
β	270.5592	269.2307	275.9040
v_0	119.8090	121.5254	120.0952
k_1	-0.3435	-0.2765	-0.1067
k_2	0.1232	0	-0.1577

TABLE IV
COMPARISON OF DISTORTION MODELS USING ODIS IMAGES IN [15]

ODIS Images			
Model	#1	#2	#3
J	840.2650	933.098	851.262
α	260.7636	258.3206	266.0861
γ	-0.2739	-0.5166	-0.3677
u_0	140.0564	137.2155	139.9177
β	255.1465	252.6869	260.3145
v_0	113.1723	115.9295	113.2417
k_1	-0.3554	-0.2752	-0.1192
k_2	0.1633	0	-0.1365

(CSOIS) at USU which has resulted in the so-called T-series of omni-directional (ODV) robots [19]. With the ODV technique, our robots including ODIS, can achieve complete control of the vehicle's orientation and motion in a plane, thus making the robots almost holonomic - hence "omni-directional". ODIS employs a novel parameterized command language for intelligent behavior generation [17]. A key feature of the ODIS control system is the use of an object recognition system that fits models to sensor data. These models are then used as input parameters to the motion and behavior control commands [16]. Fig. 1 shows the mechanical layout of the ODIS robot. The robot is 9.8 cm tall and weighs approximately 20 kgs.

B. Motivation

The motivation to do camera calibration and radial undistortion is to better serve the wireless visual servoing task for ODIS. Our goal is to align the robot to a parking lot yellow line for localization. Instead of our previous yellow line alignment methods described in [18], [20], we can align to the yellow line with a non-iterative way using a calibrated camera. The detail procedure is discussed in the next section.

C. Localization Procedure

Let us begin with a case when only ODIS's yaw and x, y positions are unknown while ODIS camera's pan/tilt angles are unchanged since calibration. The task of yellow line alignment is described in detail as follows:

• Given:

- 3D locations of yellow line's two ending points
- Observed ending points of yellow line in the image plane using ODIS camera

- ODIS camera's pan/tilt angles
- ODIS camera's intrinsic parameters
- Radial distortion model and coefficients

• Find: ODIS's actual yaw and x, y positions

Knowing that a change in ODIS's yaw angle only results in a change of angle s in the ZYZ Euler angles (a, b, s) . So, when using ZYZ Euler angles to identify ODIS camera's orientation, the first two variables a, b are unchanged. In Fig. 2, after some time of navigation, the robot thinks it is at Position 2, but actually at Position 1. Then it sees the yellow line, whose locations in 3D world reference frame are known from map (denoted by P_{A1}^w and P_{B1}^w). After extracting the corresponding points in the image plane of the yellow line's two ending points, we can calculate the undistorted image points and thus recover the 3D locations of the two ending points (denoted by P_{A2}^w and P_{B2}^w), using ODIS camera's 5 intrinsic parameters and radial distortion coefficients. From the difference between the yellow line's actual locations in map and the recovered locations, the deviation in the robot's x, y positions and yaw angle can be calculated.

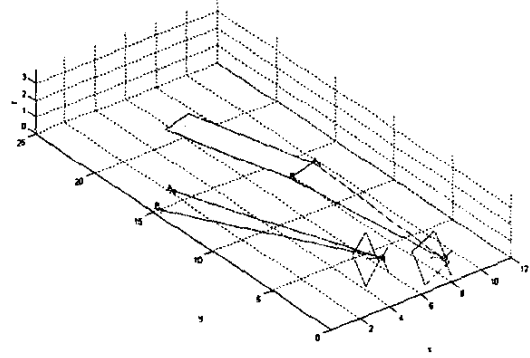


Fig. 2. The task of yellow line alignment

Let (x, y) be the undistorted points in the camera frame corresponding to the yellow line's two ending points in the 3D world frame. Let R_2 and t_2 be the rotation matrix and translation vector at position 2 (where the vehicle thinks it is at), similarly R_1 and t_1 at position 1 (the true position and orientation), we can write $R_2 = \Delta R \cdot R_1$ and $t_2 = t_1 + \Delta t$, where ΔR and Δt are the deviation in orientation and translation. If the transform from the world reference frame to the camera frame is $P^c = R^{-1}(P^w - t)$, first we can calculate P_{A2}^w and P_{B2}^w .

Let $P_{A2}^w = [X_{A2}^w, Y_{A2}^w, 0]$, we have

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = R_2^{-1} \begin{bmatrix} X_{A2}^w - t_{21} \\ Y_{A2}^w - t_{22} \\ -t_{23} \end{bmatrix}. \quad (17)$$

Since

$$\frac{X^c}{x} = \frac{Y^c}{y} = \frac{Z^c}{1}, \quad (18)$$

we have two equations containing two variables and P_{A2}^w can be calculated out. By the same way, we can get P_{B2}^w . Once P_{A2}^w and P_{B2}^w are known, we have

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = R_2^{-1} \Delta R (P_{A1}^w - t_1) = R_2^{-1} (P_{A2}^w - t_2), \quad (19)$$

where λ is a scaling factor. From (19), we get

$$R_2^{-1} [\Delta R (P_{A1}^w - t_1) - P_{A2}^w + t_2] = 0. \quad (20)$$

Similarly, we get

$$R_2^{-1} [\Delta R (P_{B1}^w - t_1) - P_{B2}^w + t_2] = 0. \quad (21)$$

Using the above two equations, we have $(P_{A2}^w - P_{B2}^w) = \Delta R (P_{A1}^w - P_{B1}^w)$, where ΔR is of the form

$$\Delta R = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 \\ \sin(\Delta\theta) & \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (22)$$

So, $\Delta\theta$ is just the rotation angle from vector $P_{A1}^w \rightarrow P_{B1}^w$ to vector $P_{A2}^w \rightarrow P_{B2}^w$. When ΔR is available, t_1 can be calculated as $t_1 = P_{A1}^w - \Delta R^{-1} (P_{A2}^w - t_2)$.

V. CONCLUDING REMARKS

This paper proposes a new radial distortion model that belongs to the polynomial approximation category. The appealing part of this distortion model is that it preserves high accuracy together with an easy analytical undistortion formula. Experiments results are presented showing that this distortion model is quite accurate and efficient especially when the actual distortion is significant. An application of the new radial distortion model is non-iterative yellow line alignment with a calibrated camera on ODIS.

REFERENCES

- [1] Emanuele Trucco and Alessandro Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.
- [2] O.D. Faugeras, Q.T. Luong, and S.J. Maybank, "Camera self-calibration: theory and experiments," in *Proceedings of the 2nd European Conference on Computer Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 321-334.
- [3] S. Bougnoux, "From projective to euclidean space under any practical situation, a criticism of self-calibration," in *Proceedings of 6th International Conference on Computer Vision*, Bombay, India, January 1998, pp. 790-796.
- [4] J. Heikkilä and O. Silvén, "A four-step camera calibration procedure with implicit image correction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 1106-1112.
- [5] P. Sturm and S. Maybank, "On plane-based camera calibration: a general algorithm, singularities, applications," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 432-437, June 1999.
- [6] Zhenyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientation," *IEEE International Conference on Computer Vision*, pp. 666-673, September 1999.
- [7] Zhenyou Zhang, "A flexible new technique for camera calibration," Microsoft Research Technical Report, <http://research.microsoft.com/~zhang/calib/>, 1998.
- [8] Juyang Weng, Paul Cohen, and Marc Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965-980, October 1992.
- [9] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323-344, August 1987.
- [10] G. Wei and S. Ma, "Implicit and explicit camera calibration: theory and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 469-480, May 1994.
- [11] Paul Ernest Debevec, *Modeling and rendering architecture from photographs*, Ph.D. thesis, Computer Science Department, University of Michigan at Ann Arbor, 1996.
- [12] Charles Lee, *Radial undistortion and calibration on an image array*, Ph.D. thesis, MIT, 2000.
- [13] Hany Farid and Alin C. Popescu, "Blind removal of lens distortion," *Journal of the Optical Society of America A, Optics, Image Science, and Vision*, vol. 18, no. 9, pp. 2072-2078, September 2001.
- [14] Zhenyou Zhang, "Experimental data and result for camera calibration," Microsoft Research Technical Report, <http://research.microsoft.com/~zhang/calib/>, 1998.
- [15] Lili Ma, "Robust flexible camera calibration: a USU implementation," CSOIS Technical Report, Department of Electrical and Computer Engineering, Utah State University, May, 2002.
- [16] Nicholas S. Flann, Kevin L. Moore, and Lili Ma, "A small mobile robot for security and inspection operations," in *IFAC Preprints of the First IFAC Conference on Telematics Applications in Automation and Robotics (TA2001)*, Weingarten, Germany, July 24-26 2001, pp. 79-84.
- [17] Nicholas S. Flann, Morgan Davidson, Jason Martin, and Kevin L. Moore, "Intelligent behavior generation strategy for autonomous vehicles using a grammar-based approach," in *Proceedings of 3rd International Conference on Field and Service Robotics FSR2001*, Helsinki University of Technology, Otaniemi, Espoo, Finland, June 11-13 2001, p. CDR0M:174.pdf.
- [18] Lili Ma, Matthew Berkemeier, Yangquan Chen, Morgan Davidson, and Vikas Bahl, "Wireless visual servoing for ODIS: an under car inspection mobile robot," in *Proceedings of the 15th IFAC Congress*, IFA, 2002, pp. 21-26.
- [19] K. L. Moore and N. S. Flann, "A six-wheeled omnidirectional autonomous mobile robot," *IEEE Control Systems*, vol. 20, no. 6, pp. 53-66, 12 2000.
- [20] Matthew Berkemeier, Morgan Davidson, Vikas Bahl, Yangquan Chen, and Lili Ma, "Visual servoing of an omnidirectional mobile robot for alignment with parking lot lines," in *Proceedings of the IEEE Int. Conf. Robotics and Automation (ICRA'02)*, IEEE, 2002, pp. 4204-4210.