Straight line feature based image distortion correction

Zhang Haofeng, Zhao Chunxia, Lu Jianfeng, Tang Zhenmin, Yang Jingyu
(Dept. of Computer Science, Nanjing Univ. of Sci. & Tech., Nanjing, Jiangsu, 210094, China)

Abstract: An image distortion correction method is proposed, which uses the straight line features. Many parallel lines of different direction from different images were extracted, and then were used to optimize the distortion parameters by nonlinear least square. The thought of step by step was added when the optimization method working. 3D world coordination is not need to know, and the method is easy to implement. The experiment result shows its high accuracy.
Key words: distortion correction; straight line feature; nonlinear least square; multi-step optimization

1 Introduction

Camera calibration consists of finding the mapping between the 3D space and the camera plane, this mapping can be separated in two different transformations: first, the displacement between the origin of 3D space and the camera coordinate system, which forms the external calibration parameters (3D rotation and translation), and second the mapping between 3D points in space and 2D points on the camera plane in the camera coordinate system, which forms the internal camera calibration parameters. Internal parameters include focal length, principle points, aspect ratio, and distortion factor.

Some classical distortion correction methods[1-3] often use the distorted pattern images to detect checkbox corners, then use them to get the internal camera calibration parameters, and at last, consider the distortion affects, optimize the distortion parameters and the other internal calibration parameters. In this paper, we propose a method which firstly corrects the distortion, and then calibrate the other internal parameters. Tsai's[1] and Zhang's[2,3] methods both optimize the distortion parameters after getting the other internal parameters, these method need to know a lot of correspondences of image points and scene points; Zhou[4] propose a method of un-calibrated lens with unknown collinear feature points, he optimized one parameter k, but the other parameters s, C_x, C_y are also important for image distortion correction, L. Iocchi[5] proposed a method optimizing the straight line features in Hough space. But it just optimized three parameters (k, C_x, C_y), and used too much proximate calculation. F. Devernay and O. Faugeras[6] provided an idea that the straight lines in scene space are all the same the straight lines in images. Then the distortion correction is to find four parameters k, C_x, C_y, s to make the lines not straight in images to be straight. Based on the idea, this paper proposed an image distortion correction method that using multi-image and step by step optimization.

2 Polynomial distortion model

The lens distortion model can be written as an infinite series:

$$\begin{cases} x_u = x_d(1 + k_1 r_d^2 + k_2 r_d^4 + \cdots) \\ y_u = y_d(1 + k_1 r_d^2 + k_2 r_d^4 + \cdots) \end{cases} \quad (1)$$

Several tests[1,7] showed that, using only the first-order radial symmetric distortion parameter k_1, one could achieve an accuracy of about 0.1 pixels. The undistorted coordinates are given by the formula:

$$\begin{cases} x_u = x_d(1 + k_1 r_d^2) \\ y_u = y_d(1 + k_1 r_d^2) \end{cases} \quad (2)$$

Where, $r_d = \sqrt{x_d^2 + y_d^2}$ is the distorted radius.

The inverse distortion model is obtained by solving the following equation for r_d,

$$r_u = r_d(1 + k_1 r_d^2) \quad (3)$$

Where, $r_u = \sqrt{x_u^2 + y_u^2}$ is the undistorted radius and r_d is the distorted radius.

This is a polynomial of degree three in r_d of the form

$$r_d^3 + c r_d + d = 0 \text{ and } c = \frac{1}{k_1}, d = -c r_u.$$

F. Devernay and O. Faugeras[6] provided a solution to solve the equation using the Cardan method, which is a direct method for solving polynomials of degree three. It has either one or three real solutions, depending on the sign of the discriminate $\Delta = Q^3 + R^2$, where $Q = c/3, R = -d/2$. If $\Delta > 0$, there is only one

Received 20 *Marth* 2007

real solution:

$$r_d = \sqrt[3]{R + \sqrt{\Delta}} + \frac{Q}{\sqrt[3]{R + \sqrt{\Delta}}} \quad (4)$$

And if $\Delta < 0$, there are three real solutions, but only one is valid because when r_u is fixed, r_d must be a continuous function of k_1. The continuity at $k_1 = 0$ gives the solution:

$$r_d = -S\cos T + S\sqrt{3}\sin T \quad (5)$$

where, $S = \sqrt[3]{\sqrt{R^2 - \Delta}}$, $T = 1/3 \arctan \sqrt{-\Delta}/R$.

Combining Eq. (2) and (3), the distorted coordinates are given by:

$$\begin{cases} x_d = x_u \dfrac{r_d}{r_u} \\ y_d = y_u \dfrac{r_d}{r_u} \end{cases} \quad (6)$$

3 Sub-pixel edge detection and extraction

Pattern with white and black parallel blocks is selected as the calibration model, illustrated in Fig. 1, because the edges of its image are easy to detect.

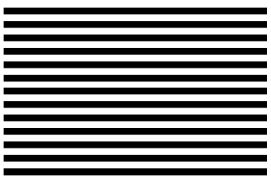


Fig. 1 Distortion calibration pattern

The first step of distortion calibration is to detect edges, sometimes the edges are less than 1 pixel, so we need to detect them with sub-pixel method. In this paper, we use the non-maxima suppression method for edge detection[8], which is the classic non-maxima suppression edge detection method with sub-pixel accuracy.

To correct the distorted images, we must extract useful edges as many as possible, which is the projection of the straight lines in real pattern. Because of the specialty of the selected pattern, the edges detected are almost useful, even if there exists some noisy edge points, we can remove them by setting a proper threshold. Following we give the algorithm of extract useful edges using point tracing:

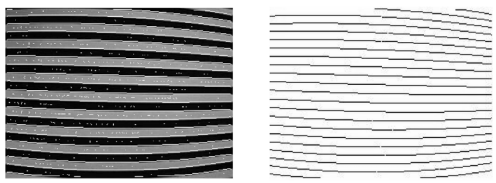
1) Select the first point of edge image as the current point;

2) Find a edge point from current point, and record it as the beginning of an useful edge, set it searched, and record it as the current point;

3) Search the 8-neighbors of the current point, if there aren't unsearched points, then this edge segment tracing over, jump to 2; If there are, add it to the current useful edge, and set it searched, and record it as the current point, loop this step.

4) Remove all the edge segments whose length is less than the threshold, and the left are the real useful edges.

The detected and extracted edges using the above method are illustrated in Fig. 2. The detected edges are shown in (a), and the extracted useful edges are in (b).



(a) Edge detection (b) Edges extraction

Fig. 2 Edge detection and extraction

4 Nonlinear least square optimization

In order to find the distortion parameters we use a measure of how much each detected edge segment is distorted. This distortion measure will then be minimized to find the best distortion calibration parameters. We chose a simple measure of distortion, which consists of doing a least squares approximation of each edge which should be a projection of a 3D segment by a line[6], and to take for the distortion error the sum of squares of the distances from the point to the line. Thus, the error is zero if the edge lies exactly on a line, and the bigger the curvature of the edge, the bigger the distortion error. We use the equation $ax + by + c = 0$ to fit the line, the equation $ax + by + c = 0$ written in matrix is

$$(x, y, 1)(a, b, c)^T = 0 \quad (7)$$

If there exists no less than 3 points, we can get the solution of (a, b, c) using linear least square, so this leads to the following expression for the distortion error of each edge segment

$$\chi = \sum_{i=1}^{n} \frac{|ax_i + by_i + c|}{\sqrt{a^2 + b^2}} \quad (8)$$

Because the pattern of this paper just contains parallel lines, we capture the pattern images of different directions to enhance the optimization. Above all, we give the distortion correction description as following:

1) Design the distortion calibration pattern as Fig. 1;

2) Capture pattern images of different directions, the images' resolution is $w \times h$;

3) Do sub-pixel edge detections on the pattern images, and extract useful edges, add them in one list, Initialize the parameters, and optimize them step by step;

4) Set $k = 0, C_x = w/2, C_y = h/2, s = 1$, and make C_x, C_y, s invariable, optimizes the parameter k using

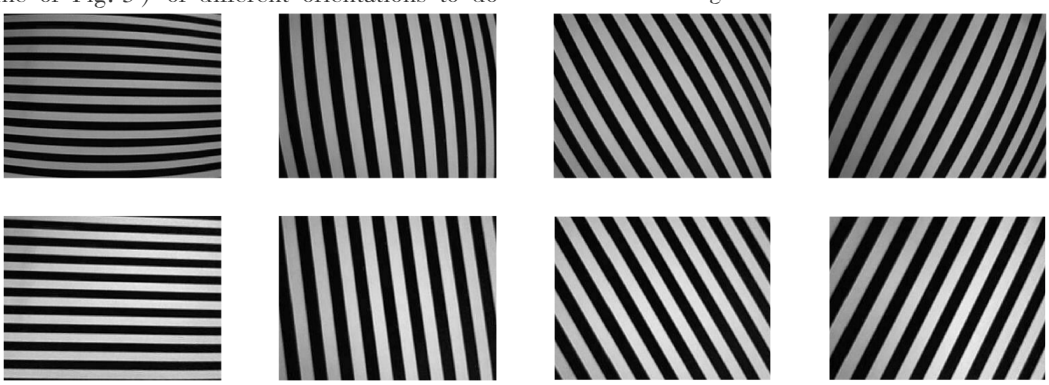
nonlinear least square (e.g. Levenberg-Mardquart);

5) Set $C_x = w/2$, $C_y = h/2$, $s = 1$, use the k got from step 4 as the initializations, and make s invariable, optimizes the parameter k, C_x, C_y using nonlinear least square;

6) Set $s = 1$, use the k, C_x, C_y got from step 5 as the initializations, optimize the parameter k, C_x, C_y, s using nonlinear least square.

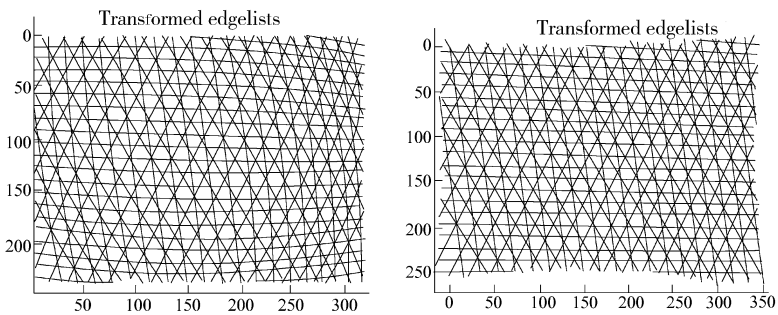
5 Experiment results and analysis

In this paper, we captured 4 images (shown in the first line of Fig. 3) of different orientations to do the distortion calibration, the corrected results are shown in the second line of Fig. 3. We extract all useful edges from the 4 images, and draw them in just one coordinate, which is shown in Fig. 4 (a), the distortion parameters are shown in Table 1, the parameters got just using single step optimization can also be seen in Table 1, we can see that step by step optimization is much better than single optimization. The corrected edges are also drawn in same coordination which is shown in Fig. 4 (b). From the corrected images, we can see that all the edges are removed the distortions, and were turned to straight lines.



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

Fig. 3 Results of image distortion correction



(a) Lines before distortion correction (b) Lines after distortion correction

Fig. 4 Comparison of before and after distortion correction

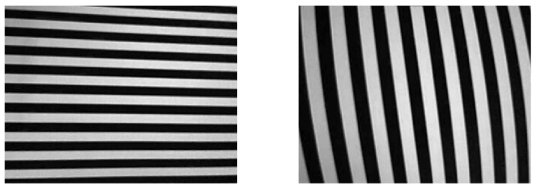
Table 1 Distortion parameters (step)

Step	k	C_x(/320)	C_y(/240)	s
1	0.135 326	0.500 000	0.500 000	1.000 000
2	0.158 625	0.439 950	0.418 456	1.000 000
3	0.156 439	0.439 931	0.401 792	0.995 522
Single	0.006 815	0.261 441	0.413 801	0.231 602

In this paper, we also give the experiment of using the parameters got from one image to correct another image. The parameters got from just one image optimization can be seen in Table 2. Fig. 5 shows the experiment, (a) is the distortion calibration image, and use the parameters to correct itself, we can see its good performance, but if we use the parameters to correct the other images, the result (shown in (b)) is rather bad, the lines become more winding.

Table 2 Distortion parameters (amount of images used)

Image	k	C_x(/320)	C_y(/240)	s
a	0.022 216	0.514 818	0.420 257	0.372 018
b	0.148 340	0.464 816	0.564 735	35.108 789
c	0.192 694	0.485 589	0.530 763	6.929 000
d	0.181 548	0.398 748	0.516 008	4.330 796
all	0.156 439	0.439 931	0.401 792	0.995 522



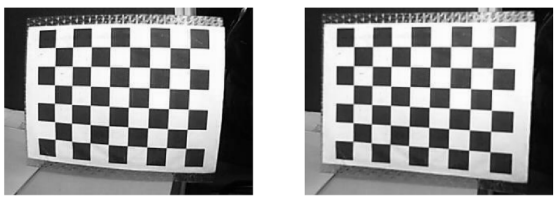
(a) Correction result of itself (b) Crrection result of the other

Fig. 5 Result of correction of just using one image optimization

We also implemented the method proposed by Zhang Zhengyou, and do comparison between it and our method. The result of Zhang's method is shown in Table 3, and the image used in Zhang's method correction result is shown in Fig. 6.

Table 3 Internal calibration parameters (Zhang's method)

α	β	γ	u_0	v_0	k_1	k_2
369.907	372.269	1.51508	138.7689	94.723 8	−0.40840	0.11550



(a) Result before correction (b) Results after correction

Fig. 6 Experiment of distortion correction

From the comparison of Table 1 and Table 3, we can find that $C_x \times w = 0.439\ 931 \times 320 = 140.778$, $C_y \times h = 0.401\ 792 \times 240$ in this paper, and $u_0 = 138.768$, $v_0 = 94.723\ 8$ in Zhang's method,

$$\alpha/\beta = [f/p_w]/[f/p_h] = p_h/p_w = 369.907/372.269 = 0.993\ 66$$

And because $s = 0.995\ 522$, we can say that the result of our method is very close to Zhang's at C_x, C_y, s. But in Zhang's method $k_1 = -0.408\ 40$, in our method $k = 0.156\ 439$. Following we will explain why $k \neq k_1$, and differ greatly from each other.

In Zhang's method, image distortion model is defined like this,

$$\begin{cases} x_d = x_u(1 + k_1 r_u^2 + k_2 r_u^4 + \cdots) \\ y_d = y_u(1 + k_1 r_u^2 + k_2 r_u^4 + \cdots) \end{cases} \quad (9)$$

If we just consider the first-order radial symmetric distortion parameter k_1, the undistorted coordinates are given by the formula

$$\begin{cases} x_d = x_u(1 + k_1 r_u^2) \\ y_d = y_u(1 + k_1 r_u^2) \end{cases} \quad (10)$$

Because of $r_d = \sqrt{x_d^2 + y_d^2}$, $r_u = \sqrt{x_u^2 + y_u^2}$, setting $k = k_o$ in our method, and $k = k_z$ in Zhang's method, the Eq. (10) can be wrote as

$$\begin{cases} r_d = r_u(1 + k_z r_u^2) \\ r_u = r_d(1 + k_o r_d^2) \end{cases} \quad (11)$$

The camera focal length we used is $f = 1.12$ mm and we normalized r_d^2 with $(w/2)^2 + (h/2)^2 = 200^2$ when we computed k_o. If we consider $r_u = 100$, $k_z = -0.408\ 40$ as known value, and get $dx = 0.003$ from camera manufacturer, then we can get $r_d = 96.25$ using Eq. (11). From equation $r_u = r_d(1 + k_0 r_d^2)$, we can get $100 = 96.25[1 + k_0(96.25/200)^2]$, so $k_0 = 0.160\ 74$. Above all, we can say that the result of our method is almost the same as the Zhang's.

6 Conclusions

In this paper, we have developed a new method to easily correct distorted images. The method only requires the camera to observe one planar pattern from a few (at least four) different orientations. The correction procedure is consists of multi-image optimization and multi-step optimization. Before optimization, we use sub-pixel non-maxima suppression method for edge detection, and line tracing for edge extraction; when executing optimization, we use polynomial distortion model and nonlinear least square to optimize the calibration parameters. The result of comparison between our method and classical method shows that our method is of good performance.

References

[1] Tsai R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-shelf TV cameral and lenses [J]. IEEE Journal of Robotics and Automation, 1987, RA-3(4): 322-344.

[2] Zhang Zhengyou. A flexible new technique for camera calibration [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330-1334.

[3] Zhang Zhengyou. Camera calibration with one dimensional objects [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(7): 892-899.

[4] Zhou Fuqiang, Hu Kun, Zhang Guangjun. Correcting distortion of camera lens with collinear points[J]. Chinese Journal of Mechanical Engineering, 2006, 42(9): 174-177. (in Chinese)

[5] Iocchi L. Design and development of cognitive robots[D]. Dipartimento di Informatica e Sistemistica: Universitù di Roma "La Sapienza". 1999.

[6] Devernay F, Faugeras O. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured environments[J]. Machine Vision and Application, 2001, 13: 14-24.

[7] Beyer H A. Accurate calibration of CCD-Cameras[A]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition[C]. Urbana Champaign, IL, 1992.

[8] Devernay, F. A non-maxima suppression method for edge detection with sub-pixel accuracy[R]. INRIA Technical Report RR-2724, 1995.

(cont. on p. 96)