PERGAMON

Pattern Recognition 34 (2001) 1199–1206

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Practical ways to calculate camera lens distortion for real-time camera calibration

Seong-Woo Park, Ki-Sang Hong*

Image Information Processing Lab., Dept. of E.E., POSTECH, San 31, Hyojadong, Namku, Pohang, Kyungbuk, 790-784, South Korea

Received 2 March 1999; received in revised form 24 April 2000; accepted 24 April 2000

Abstract

In this paper, we address practical methods for calculating camera lens distortion for real-time applications. Although the lens distortion problem can be easily ignored for constant-parameter lenses, it becomes important in the field of real-time camera calibrations, particularly for zoom lenses. Tsai's camera calibration method, which is adopted in this paper for real-time application, consists of two stages. While some camera parameters can be calculated algebraically in the first stage, a nonlinear optimization process is involved in the second stage for calculating other parameters including lens distortion, which requires a large number of calculations. However, if the lens distortion can be calculated independently of the other camera parameters, we can easily calibrate a camera with a linear method without a computational burden. We propose two different methods for calculating lens distortion independently. These methods are so simple and require so few calculations that the lens distortion can be rapidly calculated even in real-time applications. The first one uses a look-up-table (LUT) of focal length and lens distortion, which can be constructed in the initialization process. The second one is a feature-based method using the relationship between the feature points found in an image. Experiments were carried out for both methods, results of which show that the proposed methods are favorably comparable in performance with the non-real-time full optimization method. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Real-time calibration; Lens distortion; Focal length–lens distortion LUT; Collinearity; Geometric invariant; Virtual studio

1. Introduction

Today, owing to the improvement of computer technology, many real-time vision applications are being studied: among them, camera calibration. Camera calibration, in the context of three-dimensional machine vision, is a process of determining the internal geometric and optical characteristics (intrinsic parameters), and the 3D position and orientation of the camera relative to a certain world coordinate system (extrinsic parameters).

The problem of camera calibration has attracted a great deal of attention in the computer vision community in the recent three decades, and various algorithms [1–4] have been introduced. Among them, Tsai's method [4] has been most widely used. Although there have been some other approaches and applications introduced thereafter [5–9], their goals have been toward the improvement of accuracy or refinement of camera models. Little attention has been paid to the algorithmic reduction of computation time, which is critical to real-time application. This paper addresses the problem of reducing the algorithmic computation time, allowing real-time application.

There are some applications including virtual studio, where real-time camera calibration is required. Virtual studio [10,11], the main application area of this paper, is in making an object (e.g., human presenter) from a real studio look like it naturally exists in a graphic studio that is generated using computer graphics with a virtual camera observing graphic objects in graphic world. In order to make this graphic studio look like it is smoothly moving, it has to be changed at the rate of 30 frames/s.

* Corresponding author. Tel.: + 82-562-279-2216; fax: + 82-562-279-5586.

E-mail addresses: parksw@cafe.postech.ac.kr (S.-W. Park), hongks@postech.ac.kr (K.-S. Hong).

PII: S0031-3203(00)00068-6

In other words, the camera parameters of virtual camera should be changed to follow those of the real camera, which means they need to be calculated at the same rate.

In order to apply the most widely used algorithm, Tsai's algorithm, to these real-time applications it has to be modified because the nonlinear optimization involved in it due to lens distortion requires considerable computational cost. To reduce the computational burden, we found that if lens distortion can be calculated independently, the other parameters involved in the nonlinear optimization can be obtained with a linear method. The point is now how to calculate lens distortion independently.

In this paper, we propose two practical methods for calculating lens distortion independently of other parameters without burdensome calculations. One method uses a focal length-lens distortion LUT that can be constructed in the initialization process. The other method finds lens distortion using the relationship between feature points in an image without any initialization process. At this point it is important to note that the feature points used in this paper are any collinear or coplanar points.

In Section 2 we will review Tsai's camera calibration model and explain how camera parameters can be calculated with a linear method given the lens distortion. Section 3 describes our new methods for calculating lens distortion and experimental results will be presented in Section 4. Section 5 concludes this paper.

2. Tsai's calibration model

The calibration model of this paper is based on Tsai's model [4] for a set of coplanar points. Among various lens distortion factors, the first radial distortion coefficient, κ_1, is most significant and we will take it into account only. Fig. 1 illustrates Tsai's camera model.

The transformation from 3D world coordinate (x_w, y_w, z_w) to frame coordinate (X_f, Y_f) follows in the next four steps.

Step 1: Rigid-body transformation from the object world coordinate system (x_w, y_w, z_w) to the camera 3D coordinate system (x, y, z) is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T, \tag{1}$$

where R is a 3×3 rotation matrix about the world coordinate axes (x_w, y_w, z_w),

$$R = Rot(R_x)Rot(R_y)Rot(R_z) = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \tag{2}$$

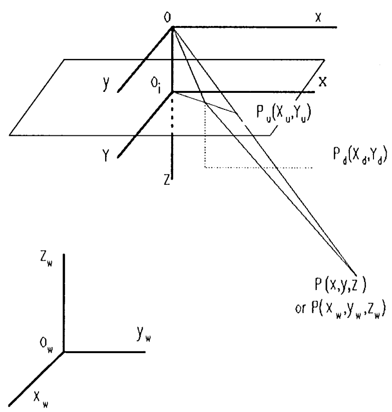

Fig. 1. Tsai's camera model.

and T is a translation vector,

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \tag{3}$$

Step 2: Projection of 3D camera coordinate to undistorted image coordinate (X_u, Y_u) is

$$X_u = f\frac{x}{z},$$

$$Y_u = f\frac{y}{z}, \tag{4}$$

where f is the effective focal length.

Step 3: Calculating the distorted image coordinate (X_d, Y_d) with a lens distortion coefficient, κ_1, is

$$X_d(1 + \kappa_1 r^2) = X_u,$$

$$Y_d(1 + \kappa_1 r^2) = Y_u, \tag{5}$$

$$r^2 = X_d^2 + Y_d^2. \tag{6}$$

Step 4: Transformation from the distorted image coordinate (X_d, Y_d) to the frame coordinate (X_f, Y_f) is

$$X_f = X_d \cdot s_x^{-1} + C_x,$$

$$Y_f = Y_d \cdot s_y^{-1} + C_y, \tag{7}$$

where the scale factor s_x, s_y, and image center (C_x, C_y) are presumed to be known. In the later experiment we use the center of expansion [12] as a constant image center which can be found in the initialization process.

Tsai used a two-stage calibration method. In the first stage he calculated the extrinsic camera parameters $[T_x, T_y, R_x, R_y, R_z]$ using RAC (radial alignment constraint). RAC represents the relationship of $\overline{O_i P_u} // \overline{O_i P_d}$, i.e., if we presume that only the lens has radial distortion, the direction of a distorted point is the same as the direction of an undistorted point. In the next stage, by minimizing the error function with f, T_z, and κ_1 as unknowns, using a standard optimization scheme such as steepest descent, he finds the optimum f, T_z, and κ_1. The error function can be defined a

$$error = \sum((X_u - X'_u)^2 + (Y_u - Y'_u)^2), \tag{8}$$

where (X_u, Y_u) is calculated from (X_f, Y_f) using Eqs. (5)–(7) and (X'_u, Y'_u) is a projected point from the world coordinate to the image using the already calculated parameters $[T_x, T_y, R_x, R_y, R_z]$, and f, T_z as a variable. This optimization process requires a large number of computations. If, in this stage, one can calculate κ_1 independently of other camera parameters, undistorted image coordinate (X_u, Y_u) can be calculated from Eqs. (5)–(7). Then, the linear equation involving f and T_z can be derived from Eqs. (1)–(4) as

$$[y_i - Y_{ui}]\begin{bmatrix} f \\ T_z \end{bmatrix} = w_i Y_{ui}, \tag{9}$$

where

$$y_i = r_4 x_{wi} + r_5 y_{wi} + r_6 z_{wi} + T_y, \tag{10}$$

$$w_i = r_7 x_{wi} + r_8 y_{wi} + r_9 z_{wi}. \tag{11}$$

Since rotation matrix R and translation T_x and T_y have all been determined at this point, y_i and w_i are fixed so that f and T_z can be calculated from Eq. (9) linearly.

3. Calculating lens distortion

3.1. LUT-based method

Lens distortion is not an important factor for constant-parameter lenses, which have a constant distortion coefficient. But zoom lenses are zoomed by a complicated combination of several lenses so that its focal length, f, and distortion coefficient, κ_1, vary during zooming operations. In this section, to calculate lens distortion, we construct a look up table (LUT) using the relationship between f and κ_1 and then refer to this LUT in real-time operations.

In the initialization process, we operate the camera from its maximum zoom-out end to maximum zoom-in end, storing the feature points which we find and identify in images. To make the $f - \kappa_1$ LUT we use the nonlinear optimization method proposed by Tsai [4]. We can find the optimum $f - \kappa_1$ LUT because the nonlinear optimization process is executed off-line. When using the coplanar pattern with small depth variation, it turns out that focal length, f, and z-translation, T_z, cannot be separated exactly and reliably even with small noise. This can be easily understood when considering that we cannot tell zooming from z-directional motion by watching input video even with human eyes. So here we use an alternative that uses T_z/f which is found to be reliable enough to use it as an index. (See Section 4.1.) Because T_z is involved in the table, T_z should be presumed to be constant. It means that the camera can move in x and y directions freely but cannot move in the z direction to use $T_z/f - \kappa_1$ LUT. Note that this holds only when coplanar pattern with small depth variation is used. When the pattern employed is three-dimensional with sufficient depth variation, $f - \kappa_1$ LUT can be used.

In the real-time process, camera parameters are calculated at the rate of 30 frames/s, so that the change of camera parameters between two adjacent frames is very small. Consequently we can use the T_z/f of the previous frame as an index for the current frame, and through iterative references we can refine them, i.e., we can refer κ_1 again using the index T_z/f that was calculated from Eq. (9) using the κ_1 referred from the T_z/f of the previous frame.

3.2. Feature-based methods

The basic idea of this section is that collinearity is not preserved under lens distortion, and projective invariants are not either. Collinearity represents a property where the line in the world coordinate is also shown as a line in an image. Projective invariants are properties between points, lines, conics, etc. These properties are preserved when the world coordinates are projected onto an image if there is no lens distortion. The basic idea of finding κ_1 from image features is a kind of searching value of κ_1 which maximally preserves the collinearity or projective invariants. In other words, to find the lens distortion for a certain frame, we first take an initial lens distortion coefficient, κ_1, and compensate frame coordinate (X_f, Y_f) using the distortion coefficient to calculate undistorted image coordinate (X_u, Y_u), then find the κ_1 that meet these properties (i.e., collinearity or projective invariant) most well between feature points in the (X_u, Y_u) coordinate. The two projective invariants tested here are the 5-coplanar points invariant and 2-line and 2-points invariant.

In this paper, we assume that there are collinear or invariant coplanar features whose projective invariant values are known in advance and that they can be detected and tracked. For the application like virtual studio where a special pattern can be employed, it is not difficult to detect and track the features. For the other applications, one should first detect such collinear or invariant coplanar features with known projective invariant values and track them thereafter.

3.2.1. Calculating lens distortion using collinearity

For a camera with lens distortion, straight lines in a pattern appear curved in an image. The dotted lines in Fig. 2 represent those distorted lines found in an image.

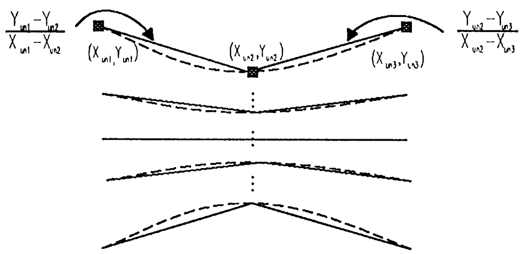

Fig. 2. Distorted lines (dotted lines) and their approximation with two lines (solid lines).

Approximation of the curved line can be done with two line segments as indicated by the solid lines in Fig. 2. Collinearity preservation means that the two line segments must be collinear if they are not distorted. In other words, their slopes must be same in the undistorted image (X_u, Y_u).

Therefore, the error function to find lens distortion is defined by the difference of the slopes, that is

$$E(\kappa_1) = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{Y_{un1} - Y_{un2}}{X_{un1} - X_{un2}} - \frac{Y_{un2} - Y_{un3}}{X_{un2} - X_{un3}}\right|^2, \tag{12}$$

where N is the number of straight lines used, and undistorted points $(X_{uni}, Y_{uni};\ i = 1, 2, 3)$ are calculated from the distorted image points (X_{dni}, Y_{dni}) that are calculated from the feature point position in the image (X_{fni}, Y_{fni}) using Eqs. (5)–(7). For vertical lines the error function we use is

$$E(\kappa_1) = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{X_{un1} - X_{un2}}{Y_{un1} - Y_{un2}} - \frac{X_{un2} - X_{un3}}{Y_{un2} - Y_{un3}}\right|^2. \tag{13}$$

Then the lens distortion coefficient minimizing the error function is what we want to obtain, i.e.,

$$\kappa_1 = \min_{\kappa_1} E(\kappa_1). \tag{14}$$

Though this method calculates lens distortion through nonlinear optimization, the amount of calculations is very small, since it is an optimization in a one-dimensional parameter space. Furthermore, when we use the κ_1 of the previous frame as an initial value of this optimization, we can minimize the number of iterations. Once the lens distortion is calculated, we can execute camera calibration using linear methods.

3.2.2. Calculating lens distortion using projective invariants

Projective invariants are preserved only if there is no lens distortion. So, for a lens having severe distortion, if we can estimate κ_1, which makes undistorted image points preserve the projective invariance, it is the lens distortion we want to find. This is just an extension of the concept used for collinearity. Based on this basic observation, we use two different invariants to find the lens distortion coefficient. One of them is the 5-coplanar point invariant and the other is the 2-line and 2-points invariant.

Five coplanar points give rise to two invariants. If we represent these points as $\mathbf{p}_i = (x_i, y_i, z_i)$, the two invariants are given by

$$I_1 = \frac{\det(m_{431})\det(m_{521})}{\det(m_{421})\det(m_{531})} \tag{15}$$

and

$$I_2 = \frac{\det(m_{421})\det(m_{532})}{\det(m_{432})\det(m_{521})}, \tag{16}$$

where m_{ijk} is a matrix composed of three points $\mathbf{p}_i$, $\mathbf{p}_j$, $\mathbf{p}_k$

$$m_{ijk} = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix}.$$

Invariant I_1 is used in this paper. Another invariant, 2-line and 2-point invariant is defined by the relationship of two lines and two points on a plane,

$$I = \frac{\mathbf{l}_1\mathbf{x}_1}{\mathbf{l}_2\mathbf{x}_1}\cdot\frac{\mathbf{l}_2\mathbf{x}_2}{\mathbf{l}_1\mathbf{x}_2}, \tag{17}$$

where $\mathbf{l}_i\mathbf{x}_j = a_i x_j + b_i y_j + c_i$, represents the algebraic distance from the point $\mathbf{x}_j$ to the line $\mathbf{l}_i$.

To find the lens distortion, we calculate an invariant I_w for the world coordinate, and I_u for the undistorted image point (X_u, Y_u), which can be calculated from the frame coordinate (X_f, Y_f) using Eqs. (4)–(7). To compare these two invariants, we select N data sets in the image. The error function for the arbitrary κ_1 is defined as

$$E(\kappa_1) = \frac{1}{N}\sum_{n=1}^{N}(I_{wn} - I_{un})^2. \tag{18}$$

The value of κ_1 minimizing this error function is what we want to find. That is,

$$\kappa_1 = \min_{\kappa_1} E(\kappa_1). \tag{19}$$

4. Experiments

In order to test the proposed method, we have designed a special grid-shaped pattern on a plate using the concept of cross-ratio [13], so that feature points can be identified automatically. Every four consecutive lines are made to have a unique cross-ratio by adjusting the spacing between lines. Fig. 3 shows a part of the pattern we designed. For the experiments we have captured more than 300 frames varying focal length of Fujinon wide lens for broadcasting camera.

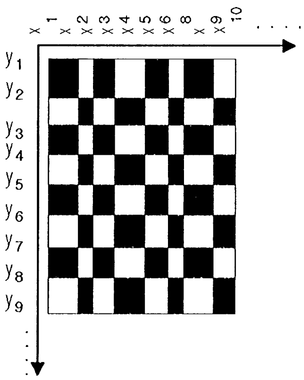

Fig. 3. A part of the pattern we designed.

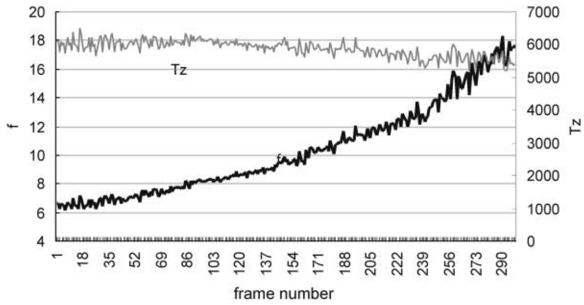

Fig. 4. f and T_z calculated by Tsai's nonlinear optimization during camera that is monotonously zooming.

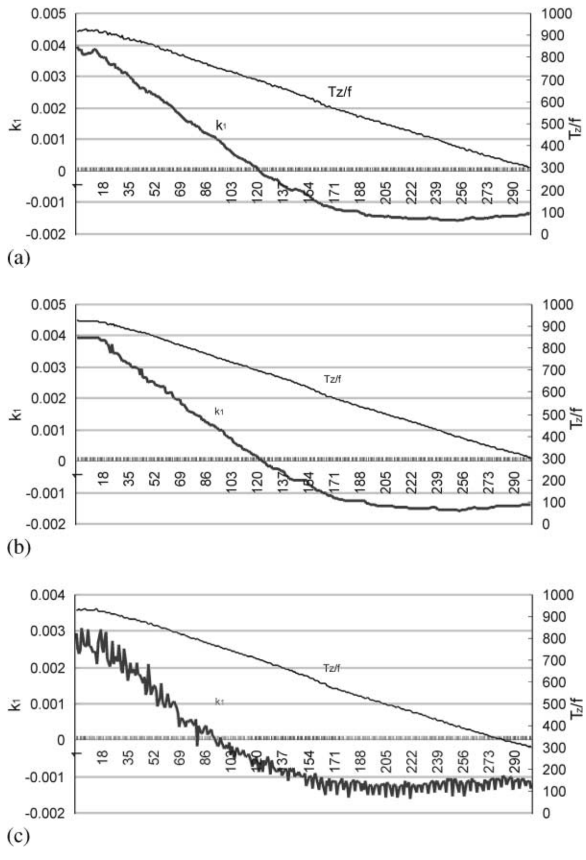

Fig. 5. T_z/f and κ_1 calculated with (a) three-variable (f, T_z, κ_1) optimization method, (b) $T_z/f - \kappa_1$ LUT-based method and (c) collinearity method. Horizontal axis represents frame number.

To identify the feature points of the pattern, we first find edges of the grid on an image using a high-pass filter, and then connect them to lines. Then, to identify the lines we calculate cross ratios of the lines we found, and compare them with the cross ratios of the pattern. After this line identification process, we calculate the intersection of the vertical lines and the horizontal lines identified. These intersections are used as the input feature points of the calibration process.

4.1. LUT-based method

Lens distortion coefficient, κ_1, varies according to the zooming operation, and the focal length uniquely represents the zooming level so that there is a unique relationship between the focal length and the lens distortion coefficient. Using this basic observation we make a LUT of the focal length, f, and the lens distortion. In real-time processes, we presume that the difference between the camera parameters of two continuous frames is not large. Therefore we can find the lens distortion coefficient by referring to the LUT using the previous f as an index of the current frame.

In the initialization process we use Tsai's nonlinear optimization method to calculate the focal length and lens distortion LUT. But in the case where a pattern with a set of coplanar points with small depth variation is used, it is very difficult to reliably separate focal length and z-directional translation, T_z, because they are strongly coupled to each other. Actually, in that case, it is not easy to tell zooming from z-directional camera motion even by the human eye. Fig. 4 shows this phenomenon. In this figure, f and T_z seem somewhat noisy, while their ratio, T_z/f, does not, as can be seen in Fig. 5.

Therefore, in this experiment, assuming a camera with fixed T_z we make a LUT using the ratio of T_z/f, instead of f, as an index of the LUT. Fig. 5(a) shows the T_z/f and κ_1 calculated in the initialization process using Tsai's optimization method. We use 300 frames for 10 s (30 frames/s). As described above, we can presume that the difference in T_z/f between two continuous frames is so small that we can find κ_1 of the current frame by looking up the $T_z/f - \kappa_1$ LUT using the previous T_z/f as an index of the current frame. As described in Section 3, for more

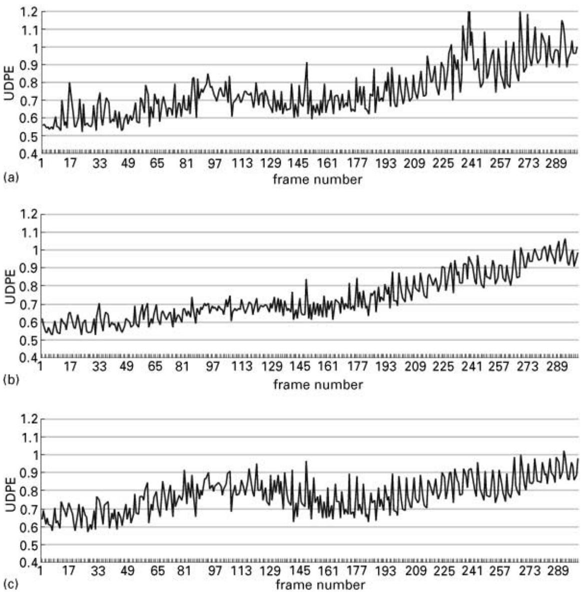

Fig. 6. UDPE of the (a) three-variable (f, T_z, κ_1) optimization method, (b) $T_z/f - \kappa_1$ LUT-based method and (c) collinearity method.

accurate lens distortion you can iterate this look-up. Fig. 5(b) shows the results of κ_1 and T_z/f calculated without an iterative look-up.

In this paper we define the undistorted projection error (UDPE) as a measure of the accuracy of the calibration. The UDPE is defined by the error between undistorted points which are projected from the world coordinate using the camera parameters calculated and the points calculated from the frame coordinate. To obtain the UDPE, we first calculate the undistorted model point (X_{um}, Y_{um}) that is calculated from the world coordinate, and another undistorted frame point (X_{uf}, Y_{uf}) from the frame coordinate (X_f, Y_f) using Eqs. (5)–(7), that is,

$$UDPE = \frac{1}{N} \sum_{n=1}^{N} ((\Delta X_u)^2 + (\Delta Y_u)^2)^{1/2}, \qquad (20)$$

where

$$\Delta X_u = (X_{um} - X_{uf})\, s_x^{-1},$$
$$\Delta Y_u = (Y_{um} - Y_{uf})\, s_y^{-1}. \qquad (21)$$

Figs. 6(a) and (b) show the UDPE's of the three variable (f, T_z, κ_1) optimization of Tsai's calibration model and the referring LUT-based method, respectively. Comparing the results, these two methods show nearly the same UDPE's. The UDPE increases as the frame number increases, because the number of feature points decreases as the camera zooms in.

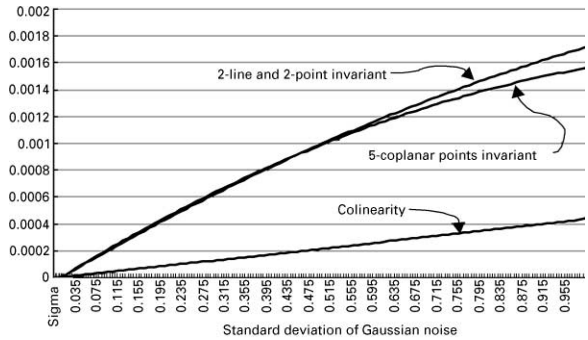

Fig. 7. The noise sensitivity of the collinearity method and two invariant methods.

4.2. Feature-based methods

In this experiment we compare the results of the three methods that use the relationship between feature points for calculating lens distortion. We first simulate noise sensitivity of these three methods, which are collinearity, 5-coplanar point invariant and 2-line and 2-point invariant. To simulate the noise sensitivity, we first project the feature points of the pattern to an image and add noise to the locations of projected feature points. To these noisy feature points we apply the three methods to find the lens distortion and calculate the RMS error of the result. Fig. 7 shows the *simulation* results of the noise sensitivity, where the abscissa and the ordinate represent the standard deviations of noise added (in pixel units) and RMS error of the lens distortion calculated, respectively. As shown in the figure, the RMS error increases as the noise increases and we can see that the noise sensitivity of the collinearity is much less than that of the two invariants. Fig. 5(c) shows the result of the experiment of the collinearity method for the continuous 300 frames. When the previous lens distortion coefficient, κ_1, is used as an initial value of the optimization process, we can find lens distortion in less than 20 iterations. Using 20 lines to find the lens distortion, we can complete the optimization process in less than 1.2 ms with a Pentium 200 MHz, while the three-variable optimization of the Tsai's model takes longer than 30 ms. Therefore it can be favorably applied to the real-time process.

Fig. 6(c) also shows the UDPE of the collinearity method. The feature points we found have an accuracy of about a half-pixel. Judging from Fig. 7, for the methods of the two invariants to have the same accuracy as the collinearity method, it seems to have to extract the feature points with an accuracy of less than 0.1 pixels.

5. Conclusion

What we have found in Tsai's algorithm, one of the most widely used for camera calibrations, is that if lens distortion can be calculated independently of the other camera parameters, calibration can be done with a linear method which saves computational cost considerably, thus enabling real-time camera calibration. In this paper we have proposed a couple of practical methods for calculating lens distortion independently. One is a look-up-table (LUT)-based method and the other is based on feature points. In the LUT method, an $f-\kappa_1$ table is constructed in the initialization stage and used in real time with the index of f calculated using the previous frame. The projective invariance method searches κ_1 through optimization, which preserves the collinearity and/or geometric invariance in the undistorted image. The geometric invariances we used are 5 coplanar points invariance and two-line two-points invariance.

Experiments were carried out with a coplanar pattern, results of which show that the performances of the LUT-based and collinearity methods are comparable to Tsai's nonlinear optimization method in terms of errors. Moreover, not to mention the LUT-based method which performs it instantly, the collinearity method achieves around a factor of 30 improvement in computation time over Tsai's nonlinear optimization method. Geometric invariances turned out to be more sensitive to noise than collinearity method. Simulation for sensitivity reveals that in order to have the same accuracy as the collinearity method, feature points should be extracted with an accuracy of less than 0.1 pixels. For a coplanar pattern, collinearity has an advantage over the LUT method in that it has no constraint on camera movement. Though tests have not been carried out for noncoplanar features, the methods proposed here, LUT and collinearity methods, can also be applied to those features.

6. Summary

The problem of camera calibration has attracted a great deal of attention in the computer vision community in the recent three decades, and various algorithms have been introduced. Among them, Tsai's method has been most widely used. Although there have been some other approaches introduced thereafter, their goals have been toward the improvement of accuracy or refinement of models. Little attention has been paid to the algorithmic reduction of computation time, which is critical to real-time applications.

Applications like camera tracking for virtual studio, the main application of this paper, require real-time camera calibration. In order to apply Tsai's algorithm to these real-time applications it has to be modified because the nonlinear optimization involved in it due to lens distortion requires considerable computational cost. To reduce the computational burden, we have found that if the lens distortion can be calculated independently of other camera parameters, calibration can be done with a linear method which would save computational cost considerably, thus enables real-time camera calibration. The point is how to obtain lens distortion parameter independently.

In this paper we propose a couple of practical methods for calculating lens distortion independently. One is look-up-table (LUT)-based method and the other is feature-based method. In the LUT-based method, the LUT made of focal length and lens distortion parameter is constructed in the initialization stage and used in a real-time operation with the index of focal length calculated at the previous frame. The feature-based method searches lens distortion parameter through optimization which preserve collinearity and/or geometric invariance in an undistorted image. The geometric invariances we use are five coplanar points invariance and two-line two-point invariance.

Experiments are carried out with a coplanar pattern, results of which show that the performances of the LUT-based and the collinearity methods are comparable to Tsai's nonlinear optimization method in terms of errors. Moreover, not to mention the LUT-based method which performs it instantly, the collinearity method achieves an improvement of around a factor of 30 in computation time over Tsai's nonlinear optimization method. The collinearity method has an advantage over the LUT method in that it has no constraint on camera movement. According to the simulation on noise sensitivity, geometric invariances turned out to be more sensitive to noise than the collinearity method. In order to have the same accuracy as the collinearity method it has to extract feature points with an accuracy of less than 0.1 pixels. Though tests have not been carried out for noncoplanar features, the methods proposed here, LUT and collinearity methods, can also be applied to those features.

References

[1] H.A. Martins, J.R. Birk, R.B. Kelly, Camera models based on data from two calibration planes, Comput. Graphics Image Process. 17 (1981) 173–180.

[2] T.M. Strat, Recovering camera parameters from a transformation matrix, Proceedings of the DARPA Image Understanding Workshop, October, 1984, pp. 264–271.

[3] A. Isaguirre, P. Pu, J. Summers, A new development in camera calibration: Calibrating a pair of mobile cameras, in: Proceedings of the International Conference of Robotics and Automation (1985) 74–79.

[4] Roger Y. Tsai, A versatile camera calibration technique for high accuracy 3-D maching vision metrology using off-the-shelf TV cameras and lenses, IEEE Journal of Robotics and Automation 3 (4) (1987) 323–344.

[5] D.B. Gennery, Camera calibration including lens distortion, Technical Report JPL D-8580, Jet Propulsion Laboratory, May, 1991.

[6] H.A. Beyer, Accurate calibration of CCD-cameras, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 1992, pp. 96–101.
[7] J. Weng, P. Cohen, M. Herniou, Camera calibration with distortion models and accuracy evaluation, IEEE Transaction on Pattern Analysis and Machine Intelligence 14 (10) (1992) 965–980.
[8] Megxiang Li, Camera calibration of the KTH head-eye system, TRITA-NA-9407, KTH, Sweden, 1996.
[9] R.G. Willson, Modeling and calibration of automated zoom lenses, CMU-RI-TR-94-03, Ph.D. Thesis, 1994.
[10] L. Blond, M. Buck, R. Galli, W. Niem, Y. Paker, W. Schmidt, G. Thomas, A virtual studio for live broadcasting: the Mona Lisa project, IEEE Multimedia 3 (1) (1996) 18–29.
[11] A. Wojdala, Challenges of virtual set technology, IEEE Multimedia 5 (1) (1998) 50–57.
[12] R.G. Willson, S.A. Shafer, What is the center of the image, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 1993, pp. 670–671.
[13] C. Coelho, A. Heller, J.L. Mundy, D.A. Forsyth, A. Zisserman, An experimental evaluation of projective invariants, in: J.L. Mundy, A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, Cambridge, 1992, pp. 87–104.

About the Author—SEONG-WOO PARK was born in Changsung, Chonnam, Korea, in 1972. He received the B.S. degree in Electronic Engineering from Chonbuk National University, Korea, in 1997 and now he is in the M.S. program at POSTECH. His research interests include computer vision, computer graphics and pattern recognition.

About the Author—KI-SANG HONG received the B.S. degree in Electronic Engineering from Seoul National University, Korea, in 1977, and the M.S. degree in Electrical & Electronic Engineering from KAIST, Korea, in 1979. He also received the Ph.D. degree from KAIST in 1984. During 1984–1986, he was a researcher in the Korea Atomic Energy Research Institute and in 1986, he joined POSTECH, Korea, where he is currently an associate professor of Electrical & Electronic Engineering. During 1988–1989, he worked in the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, as a visiting professor. His current research interests include computer vision, virtual augmented reality and pattern recognition.