Correcting Radial Lens Distortion with Advanced Outlier Elimination

Byoung-Kwang Kim, Soon-Wook Chung, Moon-Kyu Song, Woo-Jin Song
Division of Electronic and Electrical Engineering
Pohang University of Science and Technology, Republic of Korea
whitebk@postech.ac.kr, chungmal@postech.ac.kr,
demoniac@postech.ac.kr, wjsong@postech.ac.kr

Abstract

Radial lens distortion occurs when low-cost cameras or cameras with wide-angle or fish-eye lens are used. The line-based approach is based on the property that straight lines in the 3D world must be straight in 2D images; it estimates a radial lens distortion parameter by finding curved lines in the 2D image which are straight lines in 3D world. To estimate the parameter more accurately, a conventional algorithm eliminates outliers, which are defined as lines which are curved both in the 2D image and in the 3D world. We propose a novel algorithm which extends the concept of outlier to include lines which can cause errors in the estimate of the parameter even though the lines are straight in the 3D world. The proposed algorithm can eliminate outliers effectively. As a result, the proposed algorithm can estimate the distortion parameter more accurately than existing algorithms.

1. Introduction

Images taken with wide-lens or low cost camera tend to be distorted because of radial lens distortion. It can be a significant problem in 3D machine vision and image analysis which need accurate coordinates of objects in the real world. For correcting lens distortion several methods have been tried; most of them define a distortion model, estimate distortion parameters (DPs) and correct the radial distortion using the distortion model. They use different methods to estimate DPs. Some use a pattern such as a grid or a point to calculate point correspondence [1]–[3]; these methods have the disadvantage that they require a priori knowledge such as camera setting information or pre-calibrated test pattern information. To avoid these disadvantages, line-based approaches have been suggested [4]–[9]; they apply the fundamental idea that in an ideal camera such as pinhole camera, straight lines in the 3D world are straight in a 2D image. Thus, they find curved lines in an image which are considered to be straight in the 3D world, and estimate the DPs that make the curved lines straighten. These methods detect the lines in an image to estimate the parameters. Therefore, correctly detecting lines for use in estimation is very important. Actually, algorithms using the user interaction which needs an user to select or mark the lines has been proposed [8]-[9]. However, it cannot be performed automatically.

A line-based approach has been proposed [7] which use an outlier elimination procedure that considers curved lines in 2D image which are not straight lines in the 3D world. Although radial distortion characteristics cause the amount of pixel movement due to radial distortion to increase as the pixel becomes far from the center of the image, this approach applies a fixed threshold to define outliers. Therefore, it cannot correctly eliminate curved lines which cause poor estimation of DPs.

In this paper, we present a line-based approach with a more powerful outlier elimination method that expands the conventional concept to consider lines as outliers if they reduce the accuracy of DP estimation even though they are straight in the 3D world.
2. Line-based estimation with outlier elimination

2.1. Radial distortion model

Assuming that the distortion center is the center of the image, an undistorted point \( p_u = (x_u, y_u) \) can be mapped to a distorted point \( p_d = (x_d, y_d) \) as

\[
\begin{align*}
    x_u &= x_d + x_d \sum_{i=1}^{\infty} K_i r_d^{i-1}, \\
    y_u &= y_d + y_d \sum_{i=1}^{\infty} K_i r_d^{i-1},
\end{align*}
\]

where \( r_d = (x_d^2 + y_d^2)^{1/2} \) and \( K_i \) are DPs (Fig. 1). Several tests have shown that it is sufficient to use only \( K_3 \) [4]. Thus, the equations (1) and (2) can be simplified to

\[
\begin{align*}
    x_u &= x_d + x_d (K_3 r_d^2), \\
    y_u &= y_d + y_d (K_3 r_d^2)
\end{align*}
\]

If \( K_3 > 0 \), barrel distortion occurs and if \( K_3 < 0 \), pincushion distortion occurs.

2.2 Parameter estimation process

Straight lines in the 3D world must be straight in a 2D image [4]. However, radial distortion causes them to be curved. To correct the distortion using only information from a single image, curved lines in the image must be detected and used to estimate \( K_3 \). Then, the estimated \( K_3 \) is applied to a distortion model to map a distorted point \( p_d \) into an undistorted point \( p_u \). The process used in [7] to estimate \( K_3 \) consists of 4 steps, as follows.

2.2.1. Detection of points on curved line segments.

The first step detects distorted lines in the image. Curved lines are considered to consist of short line segments. Thus, the line segments are detected first. The Sobel operator is used to calculate the magnitude and angle of the segments’ gradients. Points which belong to one line segment are grouped together.

2.2.2. Linkage of curved line segments.

The second step links the line segments to make a long, curved line. The algorithm checks the distance between endpoints of two line segments, an angle of two line segments and a perpendicular distance from the endpoint of the line segment to the other line. If these measurements are less than certain thresholds, the two line segments are linked. The result is \( M \) long curved lines \( L_m (m=1, ..., M) \), each \( L_m \) consisting of \( N_m \) distorted points \( p_{d,n} (n=1, ..., N_m) \).

2.2.3. Outlier elimination.

Linkage forms long curved lines, but these lines may be either straight or curved in the 3D world. To estimate \( K_3 \) accurately, only lines that are straight in the 3D world should be used. Therefore, lines which are curved in the 3D world are defined as outliers and eliminated. The outlier elimination applies the random sample consensus (RANSAC) technique [10]. The application of the RANSAC selects one long curved line \( L_m \) randomly, then maps all \( N_m \) distorted points \( p_{d,n} = (x_{d,n}, y_{d,n}) \) of the selected line to the undistorted points \( p_{u,n} = (x_{u,n}, y_{u,n}) \) using

\[
\begin{align*}
    x_{u,n} &= x_{d,n} + x_{d,n} (K_3 r_{d,n}^2) \\
    y_{u,n} &= y_{d,n} + y_{d,n} (K_3 r_{d,n}^2)
\end{align*}
\]

where \( r_{d,n} = (x_{d,n}^2 + y_{d,n}^2)^{1/2} \) and \( K_3 \) are DPs (Fig. 1). Several tests have shown that it is sufficient to use only \( K_3 \) [4]. Thus, the equations (1) and (2) can be simplified to

\[
\begin{align*}
    x_{u,n} &= x_{d,n} + x_{d,n} (K_3 r_{d,n}^2), \\
    y_{u,n} &= y_{d,n} + y_{d,n} (K_3 r_{d,n}^2)
\end{align*}
\]

If \( K_3 > 0 \), barrel distortion occurs and if \( K_3 < 0 \), pincushion distortion occurs.

Regression is used to fit a straight line to these points. Then error \( \varepsilon_n \) from the regression line is calculated for each undistorted point and compared to a threshold \( \varepsilon_{max} \); if the percentage of points for which \( |\varepsilon_n| > \varepsilon_{max} \) exceeds a second threshold, the line is considered to be an outlier. This process is repeated a number of times. The solution with the smallest percentage of outliers is chosen.

2.2.4. Final parameter estimation
Using the solution selected in the preceding step, the distortion parameter $\kappa_3$ is estimated to minimize the cost function using only inliers.

$$\kappa_3$$

calculate the percentage of points which have $|\epsilon_n| > \epsilon_{\text{max}}$ from its associated straight line. However, for curved lines near the center, the amount of pixel movement becomes very small so

$$\kappa_3$$

Fig. 2. Distortion property according to the curve location (blue: line 1, green: line 2, red: line 3)

3. The problem with conventional outlier elimination

Lines near the image center bend less than lines near the image border even though radial distortion occurs. According to equations (1) and (2), the amount of the pixel movement due to $\kappa_3$ decreases as the distance between the line and the center decreases, even though the same $\kappa_3$ is applied. If pixel movement is small, the difference between the curved line and regression line is also small so that when $\kappa_3$ is estimated using distorted lines near the center, it can differ greatly from the true value of $\kappa_3$. Thus, curved lines near the center can cause an inaccurate estimation of $\kappa_3$, even though they are straight lines in the 3D world. The outlier elimination method in [7] uses a fixed threshold $\epsilon_{\text{max}}$ to

$$\kappa_3$$

they tend to be classified as inliers because they have a smaller percentage of points with $|\epsilon_n| > \epsilon_{\text{max}}$ at a given $\kappa_3$ than lines farther from the center. As an example (Fig. 2), if a line is classified as an outlier when more than 20% of points have $|\epsilon_n| > \epsilon_{\text{max}}$, line 3 is classified as an inlier over a wider range of $\kappa_3$ than is line 2. This is because, compared to line 2, line 3 is nearer the center, so the amount of pixel movement according to $\kappa_3$ is smaller and most points do not have $|\epsilon_n| > \epsilon_{\text{max}}$. Therefore, the error range of the estimate $\kappa_3$ increases as the distance from the line to the center decreases. This means that the estimated $\kappa_3$ obtained using lines near the center are very sensitive to $\epsilon_{\text{max}}$ and can cause the estimate to be incorrect.

Lines close to the image center can also cause incorrect estimation of $\kappa_3$ in real images (Fig. 3). At a

Fig. 3. Distortion property according to the curve location (blue: line 1, green: line 2, red: line 3)
given $\kappa_3$, lines near the image center (Fig. 3(a)) have a smaller percentage of undistorted points with $|\epsilon_n| > \epsilon_{\text{max}}$ than lines far from the center (Fig. 3(b)). If a line is classified as an outlier when more than 20% of points have $|\epsilon_n| > \epsilon_{\text{max}}$, line 3 have widest range considered as the inlier among lines. Estimation from the line 3 can have most error, and can reduce estimation accuracy even though line 3 is straight in the 3D world.

**4. Proposed algorithm**

The proposed algorithm defines outliers in a broader sense to include lines which reduce the accuracy of the $\kappa_3$ estimation even if they are straight in the 3D world. Most of these outliers are near the image center where the amount of pixel movement due to distortion is small. Therefore, we must apply a strict threshold of estimation to narrow the range of $\kappa_3$ over which the line is classified as an inlier. The distance between the image center and $L_m$ is defined as the length of a perpendicular line from the image center to the linear regression line for $L_m$ (Fig. 4). The weight is normalized; it is 0 when $d_m$ is 0, and 1 when $d_m$ is the distance between the center and vertex of the image. The weighted threshold $\epsilon_{m,\text{max}}$ is expressed as

$$\epsilon_{m,\text{max}} = \frac{2d_m}{w^2 + h^2} \delta_{\text{max}}, \text{ for } m = 1, \ldots, M \tag{10}$$

where $\delta_{\text{max}}$ is a user parameter, $w$ is image width and $h$ is image height.

Equation (10) was applied to the lines in Fig. 2(a). The percentage of points with $|\epsilon_n| > \epsilon_{m,\text{max}}$ increased for all lines (Fig. 5). If a line is classified as an inlier when less than 20% of points have $|\epsilon_n| > \epsilon_{m,\text{max}}$, line 3 is an outlier over the entire range of $\kappa_3$ values. The ranges over which line 2 and line 3 are classified as inliers is narrower than in Fig. 2; i.e., applying the weighted threshold according to $d_m$ can narrow the range of $\kappa_3$ over which a line is classified as an inlier. As a result, the inliers include only lines which can help to estimate $\kappa_3$ accurately.

Equation (10) was also applied to the lines in Fig. 3(a). The percentage of points with $|\epsilon_n| > \epsilon_{\text{max}}$ increased all lines (Fig. 6). Line 1 is far from the center, so this line is still an inlier over a narrow range of $\kappa_3$, even if $\epsilon_{\text{max}}$ is used as the error threshold. Line 2 is farther from the center than line 1, so applying $\epsilon_{\text{max}}$ narrowed the range of $\kappa_3$ over which line 1 is an inlier, and made the estimated $\kappa_3$ more reliable. Line 3 is nearest the center; in this line, more than 20% of points had $|\epsilon_n| > \epsilon_{\text{max}}$, and was classified as an outlier, over the whole range of estimated $\kappa_3$. In this way, because the lines near the image center can have a higher percentage of points with $|\epsilon_n| > \epsilon_{\text{max}}$ than

Fig. 4. The distance $d_m$ between the image center and the curve

Fig. 5. The percentage of points in Fig. 2(a) with $|\epsilon_n| > \epsilon_{\text{max}}$ and $|\epsilon_n| > \epsilon_{m,\text{max}}$ for $\kappa_3$.

Fig. 6. The percentage of points in Fig. 3(a) with $|\epsilon_n| > \epsilon_{\text{max}}$ and $|\epsilon_n| > \epsilon_{m,\text{max}}$ for $\kappa_3$. 

lines farther from the center, the proposed algorithm applies a stricter threshold for outlier elimination.

5. Experiments

Fig. 7. The result image (green : inlier, red : outlier)

Fig. 8. The result image (green : inlier, red : outlier)
We compared the proposed algorithm to [7] using a natural image (Fig. 7). Because [7] applied a fixed threshold to perform outlier elimination, most of the curved lines near the center were detected as inliers. However, the remaining inliers caused 3 to be estimated incorrectly. As a result, the image was not completely corrected (Fig. 7(b)). The proposed algorithm applied the weighted threshold to detect curved lines near the image center as outliers. As a result, 3 was estimated correctly and correction was good (Fig. 7(c)). Both algorithms were also applied to another image, and the proposed algorithm had good performance (Fig. 8).

Fig. 9 shows the relative frequency of 3 among inliers with outlier elimination of [7] and the proposed algorithm, respectively. In the experiment, 3 is estimated from each inlier line. The range of the estimated 3 was much greater using [7] (Fig. 9(b)) than when using the proposed algorithm (Fig. 9(c)). This demonstrates that the proposed algorithm provides a narrow range of the estimated 3, which are sufficiently accurate to correct even complicated images (Fig. 8).

6. Conclusion

We present a method of correcting radial lens distortion that uses a weighted error threshold to eliminate outliers in images. The proposed method estimates the distortion parameter 3 using only a single image without any a priori knowledge. The proposed method identifies two kinds of outliers: lines that are curved in the 3D world; and lines that are straight in the 3D world but which are compatible with a wide range of 3 values, and can thus reduce the accuracy of 3 estimation. As a result, since the range of the 3 considered to be inliers was reduced, estimation accuracy was increased, and correction performance was improved.

Acknowledgement

This work was supported in part by the IT R&D program of MKE/MCST/IITA (2008-F-031-01, Development of Computational Photography Technologies for Image and Video Contents), in part by the Brain Korea (BK) 21 Program funded by MEST, and in part by the HY-SDR Research Center at Hanyang University under the ITRC Program of MKE, Korea.

References


Fig. 9. Relative frequency of 3 in Fig. 8


