Nonmetric Calibration of Camera Lens Distortion: Differential Methods and Robust Estimation

Moumen Ahmed, Member, IEEE, and Aly Farag, Senior Member, IEEE

Abstract—This paper addresses the problem of calibrating camera lens distortion, which can be significant in medium to wide angle lenses. Our approach is based on the analysis of distorted images of straight lines. We derive new distortion measures that can be optimized using nonlinear search techniques to find the best distortion parameters that straighten these lines. Unlike the other existing approaches, we also provide fast, closed-form solutions to the distortion coefficients. We prove that including both the distortion center and the decentering coefficients in the nonlinear optimization step may lead to instability of the estimation algorithm. Our approach provides a way to get around this, and, at the same time, it reduces the search space of the calibration problem without sacrificing the accuracy and produces more stable and noise-robust results. In addition, while almost all existing nonmetric distortion calibration methods need user involvement in one form or another, we present a robust approach to distortion calibration based on the least-median-of-squares estimator. Our approach is, thus, able to proceed in a fully automatic manner while being less sensitive to erroneous input data such as image curves that are mistakenly considered projections of three-dimensional linear segments. Experiments to evaluate the performance of this approach on synthetic and real data are reported.

Index Terms—Camera calibration, lens distortion, robust estimators.

I. INTRODUCTION

Calibration is an important component of any vision task which seeks to extract geometric information from a scene. Being linear if expressed in terms of projective geometry, the ideal pinhole model simplifies a lot of considerations on geometry in which cameras are involved. However, for some applications which require high accuracy, or in cases where low-cost or wide-angle lenses are used, the pinhole model is not sufficient and more parameters should be estimated to take into account camera lens distortion. When the lens has a nonnegligible distortion, using the ideal, distortion-free model may result in high calibration error [1].

The distortion parameters are often estimated along with all (extrinsic and intrinsic) parameters of the camera model (see, for example, [2], [3]). This is done using a set of three-dimensional (3-D) to two-dimensional (2-D) correspondences extracted with the help of a calibration object of known structure. The problem with these methods is the fact that there is some kind of coupling between internal parameters, including distortion parameters, and external parameters that result in high errors on the camera internal parameters [3]. Moreover, obtaining accurate coordinates of 3-D scene points is sometimes demanding or impossible (e.g., in case of video recordings, e.g., surveillance video recordings).

In contrast, another family of nonmetric methods have been proposed, which do not rely on known scene points or need calibration objects of known structure. Instead, these methods take advantage of geometric invariants of some image features, e.g., straight lines [4]–[8], vanishing points [9], or the image of a sphere [10]. The methods proposed in [4]–[8] rely on the fact that straight lines in the scene must always project to straight lines in the image. Becker and Dove [9] used three mutually orthogonal sets of parallel lines and the minimum vanishing point dispersion constraint to recover distortion parameters. However, their constraint of the existence of triplets of orthogonal lines is less likely to hold in urban settings. Some proposed methods have made use of point correspondences between multiple views, e.g., [11]–[13]. These methods rely on establishing the correspondence between image points from different views, which is not easy to solve and is likely to produce some false data to the distortion algorithm. Other approaches to estimating distortion parameters were also proposed for image and video sequences [14], [15].

In this paper, we propose two nonmetric methods for lens distortion calibration. We derive two new distortion measures that can be optimized using nonlinear search techniques to find the best distortion parameters that straighten the image lines. Similar to the other line straightness based methods [4]–[8], our methods thus share the advantage of making very few assumptions on the observed world; All they need is a scene image containing 3-D segments, which is easy to find in indoor scenes or city scenes. However unlike those, to alleviate probable problems of nonlinear optimization, we also show how to use these measures to find fast, analytic solutions for the distortion coefficients.

It has been reported [4], [8] that including both the distortion center and the decentering coefficients in the nonlinear optimization may lead to instability of the estimation algorithm. To avoid this, some researchers (e.g., [8]) used as a coarse-to-fine exhaustive search for the distortion center around the image center; however, this has resulted in a prolonged search time. Here, we prove that the deviation of the distortion center from its true location under both lens radial and decentering distortion is equivalent to adding two additional decentering distortion:

Manuscript received May 15, 2002; revised March 17, 2004. This work was supported by the U.S. Army under Grant DABT60-02-P-0063, Perception for Autonomous Mobility, Phase I and Phase II. The associate editor coordinating the review of this manuscript and approving its publication was Prof. Patrick Bouhtoum.

M. Ahmed is with the Electrical Engineering Department, Assiut University, Assiut 71516, Egypt (e-mail: moumen@aun.edu.eg).
A. Farag is with the Computer Vision and Image Processing Laboratory, University of Louisville, Louisville, KY 40292 USA (e-mail: farag@cvip.uofl.edu).

Digital Object Identifier 10.1109/TIP.2005.846025
tion terms. This enables us to fix the distortion center at an appropriate location (e.g., image center) and then use the two centering distortion coefficients to compensate for reasonable deviations of the center from the true location. This reduces the search space of the calibration problem without sacrificing the accuracy and produces more stable and noise-robust results.

In almost all existing nonmetric distortion calibration methods [4], [5], [7], [8], some user involvement for data preparation is needed in one form or another. For example, the user should manually select the image curves that correspond to scene linear segments [5], [8]). We propose to use a robust approach based on the least-median-of-squares (LMedS) estimator to discard outliers in the data that might enter the estimation algorithm in several forms. The LMedS method can theoretically handle up to 50% outliers in the input data [16]. Our approach is thus able to proceed in a fully automatic manner while being less sensitive to erroneous data such as image curves that are mistakenly considered as projections of 3-D linear segments.

This paper is organized as follows. Section II describes the camera distortion model. Section III derives a distortion measure in order to straighten the distorted image lines, followed by analytic and nonlinear techniques to minimize it in Section IV. A similar measure of lens distortion that is based on image gradient is proposed in Section V. Section VI proposes a robust calibration procedure based on the LMedS estimator. Some experimental results are reported on both real and synthetic data in Section VII, followed by our concluding remarks in Section VIII.

II. LENS DISTORTION MODEL

The two principal forms of distortion considered in videometry and photogrammetry applications are radial and decentering (also called tangential) distortion. The standard model for the radial and decentering distortion [17] is mapping $\mathcal{U}$ from the distorted image coordinates $(x^d, y^d)$ that are observable, to the undistorted image coordinates $(x^n, y^n)$, which are not physically measurable, according to

$$\mathcal{U}: (x^d, y^d) \rightarrow (x^n, y^n)$$

such that

$$x^n = x^d + \bar{x}^d (\kappa_1 r^2_d + \kappa_2 r^4_d + \kappa_3 r^6_d + \ldots) + \left[ p_1 \left( r^2_d + 2x^d \bar{y}^d \right) + 2p_2 x^d y^d \bar{y}^d \right] [1 + p_3 r^2_d + \ldots]$$

$$y^n = y^d + \bar{y}^d (\kappa_1 r^2_d + \kappa_2 r^4_d + \kappa_3 r^6_d + \ldots) + \left[ p_2 \left( r^2_d + 2y^d \bar{x}^d \right) + 2p_1 y^d x^d \bar{x}^d \right] [1 + p_3 r^2_d + \ldots]$$

(1)

where

$$\bar{x}^d = x^d - c_x, \quad \bar{y}^d = y^d - c_y, \quad r^2_d = \bar{x}^2 + \bar{y}^2$$

and $\kappa_1, \kappa_2, \kappa_3, \ldots$ are the coefficients of radial distortion and $p_1, p_2, p_3, \ldots$ are the coefficients of the decentering distortion.

III. LINE STRAIGHTNESS METHOD

The goal of the distortion calibration is to find the transformation that maps the actual camera image plane onto an image following the perspective camera model. To find the distortion parameters, the following fundamental property is often used: A camera follows the perspective camera model if and only if the projection of every 3-D line in space onto the camera plane is a line. Consequently, all one needs is a way to find projections of 3-D lines in the image, and a way to measure how much each line is distorted in the image. This distortion measure will then be minimized to find the best calibration parameters. Several distortion measures could be used, e.g., the mean curvature of the line points or the distance between the line joining the imaged line’s ends and its mid-point. One common such measure is the sum of squared distances of the edge points from the straight lines on which they should lie [4], [6], [7]. Let $(x^n_{i,1}, y^n_{i,1})$ denote the undistorted point computed from the $i$th point on the curved line $l$, $(x^d_{i,1}, y^d_{i,1})$ according to the distortion model $\mathcal{U}$. Let the number of curved lines be $L$ and the number of points on each line be $N_l$. Then, the distortion measure is given by

$$\xi_i = \sum_{l=1}^{L} \sum_{i=1}^{N_l} \left[ x^n_{i,1} \sin(\theta_l) + y^n_{i,1} \cos(\theta_l) - \eta_l \right]^2$$

(2)

where $\theta_l$ and $\eta_l$ are the best-fit line parameters of the curve $l$ after applying the mapping $\mathcal{U}$.

These measures lead to nonlinear objective functions that need efficient search algorithms. In what follows, we derive a new distortion measure that can be minimized by nonlinear optimization algorithms and from which, closed-form solutions can also be derived to solve for the distortion parameters.
A. Proposed Distortion Measure

Our proposed measure employs the fundamental property stated above. Suppose we have a line \( l \) in the undistorted image plane. Every point \((x^u, y^u)\) on the line satisfies the equation
\[
a x^u + b y^u + c = 0
\]
where \( a, b, \) and \( c \) are constants for the specific line \( l \), with \(-a/b\) being the line slope. Each point on the line is related to a point \((x^d, y^d)\) in the distorted image plane according to (1). This means that both coordinates of the line point are functions of \((x^d, y^d)\). Accordingly, the last equation can be written as
\[
f(x^d, y^d) = ax^u(x^d, y^d) + by^u(x^d, y^d) + c = 0
\]
where \( f(x^d, y^d) \) describes the equation of the corresponding curve in the distorted image plane. The elemental change in \( f \) at any distorted image point \((x^d, y^d)\) can be expressed as
\[
\delta f = a \left[ \frac{\partial x^u}{\partial x^d} \delta x^d + \frac{\partial x^u}{\partial y^d} \delta y^d \right] + b \left[ \frac{\partial y^u}{\partial x^d} \delta x^d + \frac{\partial y^u}{\partial y^d} \delta y^d \right] = 0
\]
where all the four partial derivatives can be directly computed from (1) (for an example, see [22]). Hence, one can see that the slope of the line \( s \) in the undistorted plane (which should equal \(-a/b\)) is related to the slope of the tangent \( \delta y^d/\delta x^d \) to the curve at point \((x^d, y^d)\) by
\[
s(x^d, y^d) = \frac{\partial y^u}{\partial x^d} + \frac{\partial y^u}{\partial y^d} \frac{\delta y^d}{\delta x^d}.
\]

In the problem of distortion calibration, we usually have a number of distorted points in the image plane. Under the correct values of the distortion parameters, the slopes computed from the last equation at all these points should be the same if the points are to lie on the same line in the undistorted image. Therefore, we can define the following distortion measure. Given a number of edge points, \((x_i^d, y_i^d), \ i = 1, \ldots, N\), that should belong to the same line in the undistorted image, we can compute the slopes of the tangents at these points, and, hence, we can solve for the distortion parameters that minimize the error\(^1\)
\[
\xi_s = \sum_{i=2}^N \left( s(x_i^d, y_i^d) - s(x_{i-1}^d, y_{i-1}^d) \right)^2.
\]

Clearly, this measure would be zero if the points are mapped to a perfect linear segment in the undistorted image and the more the segment would be distorted, the bigger the measure. To improve the accuracy, several curves distributed through the image ought to be used. In this case, the sum of the error in (7) for all the segments is employed.

The slope of the chain curve at any point is estimated by approximating the curve points within a window of size \(2W+1\) (in our implementation, \( W = 3 \)) centered at the point by a polynomial (we used one of third order). Then, the slope can be easily estimated from the fitted polynomial. The window free parameter \( W \) could be increased to alleviate noise effects (at the expense of possible loss of some fine details of the curve slopes).

To give an idea on how the distortion measure in (7) is behaving as a function of some distortion coefficients, Fig. 1 plots the measure as a function of the radial distortion \( \kappa_1 \). The error function \( \xi_s \) is computed for ten synthetic lines distributed over a \( 512 \times 512 \) image, distorted by \( \kappa_1 = 2 \times 10^{-3} \) and corrupted by zero-mean Gaussian noise of standard deviation 0.1 pixels. The function is smooth, qualitatively parabolic in shape, and with one local (global) minimum at the correct distortion values.

IV. Distortion Calibration

To calibrate the distortion parameters, the distortion measure in (7) can be minimized using nonlinear optimization algorithms, e.g., the well-known Levenberg–Marquardt (LM) algorithm. Such algorithms must be supplied with an appropriate starting point. The distortion center \((c_x, c_y)\) is often set initially at the center of the image, while zero values are assumed for the distortion coefficients, i.e., the system is initially assumed distortion-free. In our experiments, however, to help the minimization algorithm take off, we initialized\(^2\) \( \kappa_1 \), such that \(|\kappa_1| \times r^3 \leq 0.5\), where \( r \) is the distance from the center to the corner of the image and the sign of \( \kappa_1 \) being positive (negative) for pincushion (barrel) distortion, while the other parameters are initialized with zeros. Moreover, it is better [12] to assume initially that there are fewer distortion coefficients (e.g., only \( \kappa_1 \)) that matter. After finding the best values for those parameters, one can use these values as a starting guess and try searching for other parameters as well.

Although nonlinear optimization algorithms can achieve high accuracy, they may suffer from slow convergence, instability, and local minima and may end up with a false solution if a good initial guess is not available. It will be, thus, advantageous to devise analytic, closed-form solutions for the distortion parameters. Having advantages of speed and simplicity, closed-form solutions can also provide the required starting points for the more-accurate nonlinear optimization

\(^1\)A more general form would be \( \sum_{i=1}^N \sum_{j=1}^N (s(x_i^d, y_i^d) - s(x_j^d, y_j^d))^2 \).

\(^2\)From the assumption that distortion would result in a pixel deviation of 0.5 at image corners.
algorithms. Unlike earlier approaches to lens-distortion calibration (e.g., [4] and [6]–[8]), we here use the methodology of the last section to devise closed-form solutions for all the distortion coefficients.

To provide these solutions, we at first make use of the assumption that a good estimate of the distortion center is known. This assumption is not very restricting (as it might appear at first). Some earlier researchers (e.g., [5]) fixed the distortion center at the center of the image. Other researches [2], [3] assumed the distortion center coincided with the camera principal point, which could be independently estimated without full camera model calibration using some other existing methods [23] or by the autocollimated laser technique [21]. This situation (known distortion center) is also applicable if solving for the distortion coefficients is nested within a coarse-to-fine search for the distortion center (as is the case in [8]). In such a case, the closed-form solution will provide a quick solution for the distortion coefficients at each tested location of the distortion center. In the following, we use this assumption to derive closed form solutions. Then we show how this assumption can be relaxed.

A. Linear Closed-Form Solution

The reason behind the existence of closed-form solutions for our method comes from the fact that (6) will become a rational function of two linear polynomials in the distortion coefficients, once the distortion center is known. One way to derive those closed-form solutions if the number of unknown coefficients ≤ 2 is to form a set of simultaneous quadratic polynomials in the unknowns [24], which can be solved by finding the common polynomial root [25]. Below, we present a simpler, more general solution.

At each point of the curves extracted from the distorted image, (6) can be applied, with the left-hand side (LHS) being the slope of the undistorted line to which these points belong. That slope can be estimated from the curve points by least-square linear regression. With the LHS of (6) being known, each point will thus yield one linear equation in the distortion coefficients. All the equations obtained from all points are stacked in the form $A\mathbf{x} = \mathbf{b}$, where $A$ is a $N \times L \times p$ matrix, with $N$ being the average number of points used from each chain and $L$ is the number of curve; $\mathbf{x}$ is a vector of the unknown distortion coefficients and $\mathbf{b}$ is a vector of known quantities of length $N \cdot L$. This over-determined set of equations can be efficiently solved using singular value decomposition. For example, if we are to estimate only $\kappa_1$ and $p_1$, the equations will be in the form

$$
A = \begin{bmatrix}
   a_{11} & a_{12} \\
   \vdots & \ddots \\
   a_{21} & a_{22}
\end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \kappa_1 \\ p_1 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \hat{s}_1 - s_{1,1} \\ \vdots \\ \hat{s}_2 - s_{2,1} \end{bmatrix}
$$

(8)

with

$$
\begin{align*}
a_{11} &= 2x_{1,1}y_{1,1} + s_{1,1} (3y_{1,1}^2 + x_{1,1}^2) - \hat{s}_1 (3x_{1,1}^2 + y_{1,1}^2 + 2s_{1,1}x_{1,1}y_{1,1}) \\
a_{12} &= 2y_{1,1} + 2s_{1,1}x_{1,1} - \hat{s}_1 (6x_{1,1} + 2s_{1,1}y_{1,1}), \ldots \\
a_{21} &= 2x_{2,1}y_{2,1} + s_{2,1} (3y_{2,1}^2 + x_{2,1}^2) - \hat{s}_2 (3x_{2,1}^2 + y_{2,1}^2 + 2s_{2,1}x_{2,1}y_{2,1}) \\
a_{22} &= 2y_{2,1} + 2s_{2,1}x_{2,1} - \hat{s}_2 (6x_{2,1} + 2s_{2,1}y_{2,1}), \ldots
\end{align*}
$$

where $(x_{i,j}, y_{i,j})$ denotes the $i$th distorted point on chain $l$ with tangent slope $s_{i,j}$; $\hat{s}_i$ is the estimated slope of the line corresponding to chain $l$.

It is very important to note that the LHS of (6) is the slope of the ideal undistorted line, which is typically unknown since all the given data are in the distorted plane. The reason why this approach works is that it makes use of the more reliable slope information of the extracted segments, which is not considerably affected by the radial and decentering distortion acting on the curve points. Accordingly, the slope obtained from best-fit line of the distorted curve points is usually close to the “unknown” slope of the undistorted line. This is in contrast to the intercept component of the best-fit line, which largely varies across the distortion parameters. Fig. 2 illustrates the changes of the slope (represented as angles in degrees) in (a) and the intercept in (b) of the best-fit line as a function of the radial distortion $\kappa_1$. This figure was obtained for ten synthetic lines of various orientations and locations in a 512 × 512 image distorted by different
values of \(k_1\) up to \(10^{-5}\). The magnitude of this \(k_1\) range includes distortions far stronger than those found in wide-angle imaging sensors [8]. Moreover, the lines were selected to be near the boundaries of the image with the distortion center being at the image center so that the distortion has a prominent effect on the lines. Across this range, the slopes of the lines showed much smaller variation from the true slopes (shown in circles) compared to that of the intercept. Similar results and observations are also obtained for the ten lines in Fig. 3 across a wide magnitude range of the decentering distortion \(p_1\).

Moreover, one can improve further the first estimates of the lines slopes used to solve for the distortion parameters by undistorting the points by the obtained distortion coefficients and then estimating the slopes again. Afterward, the linear approach is run for a second time.

### B. Uncertainty of the Distortion Center

In the previous subsection, we made use of the assumption that the distortion center is known in order to obtain analytic solution for the distortion coefficients. Here, we relax this assumption by allowing for uncertainty in the location of the center. In order to do this, we investigate the effect of fixing the distortion center at a location different from the true one on the distortion model. We, thus, need to know how much accuracy is lost when using another model instead of the true one. Comparing the distortion parameters of the two models does not give a meaningful, quantifying difference measure. We, therefore, measure how close (or different) two distortion models are in terms of their effect on an image. In the following, we define a closeness measure between two models, then give two lemmas (for proof, see Appendix).

Given a number of points \(\{(x_i, y_i)\}\) distributed on a distorted image (ideally all image points), let \(\{(x_i(\mathcal{U}), y_i(\mathcal{U}))\}\) be these points undistorted using the model \(\mathcal{U}\). We measure the closeness between the model \(\mathcal{U}\) and another distortion model \(\mathcal{U}'\) as

\[
\mathcal{C}(\mathcal{U}, \mathcal{U}') = \max \left( \max_i \left| x_i(\mathcal{U}) - x_i(\mathcal{U}') \right| \right) \max_i \left| y_i(\mathcal{U}) - y_i(\mathcal{U}') \right|. \tag{9}
\]

\(\mathcal{C}(\mathcal{U}, \mathcal{U}')\) uses the uniform\(^{3}\) norm to measure the maximum absolute pixel deviation in either coordinate of the image points after being undistorted by the two models. In other words, one can say that \(\mathcal{U}'\) approximates \(\mathcal{U}\) on these image points within error \(\mathcal{C}(\mathcal{U}, \mathcal{U}')\).

**Definition 1:** A distortion coefficient is considered to be dominant in a distortion model if the closeness measure of (9) on an image between the model with all distortion coefficients and the model with only that coefficient is less than the precision of image point localization and measurement.

One can, thus, neglect the effect of those other coefficients in the model on the image.

**Lemma 1:** Under lens, radial distortion with dominant \(k_3\), shifting the distortion center \((c_x, c_y)\) by \((\Delta_x, \Delta_y)\) approximates adding two decentering distortion terms of \(-k_1\Delta_x\) and \(-k_1\Delta_y\) to the model, with an approximation error throughout a \(W \times H\) image bounded above by \(|k_1||2W + H|\Delta_x^2 + 2\Delta_y^2|\), where \(\Delta = \max(|\Delta_x|, |\Delta_y|)\).

**Lemma 2:** For a decentering distortion model with \(p_1\) and \(p_2\), shifting the distortion center \((c_x, c_y)\) by \((\Delta_x, \Delta_y)\) yields a distortion model that approximates the original model throughout a \(W \times H\) image with an approximation error bounded above by

\[
\max \left( 2(2|p_1| + |p_2|) \Delta^2 + (3W + H)|p_1| + (W + H)|p_2| \right) \Delta,
\]

\[
2(|p_1| + 2|p_2|) \Delta^2 + (W + H)|p_1| + (W + 3H)|p_2| \Delta
\]

where \(\Delta = \max(|\Delta_x|, |\Delta_y|)\).

The two lemmas prove that shifting the distortion center from its true location, under both radial and decentering distortion, adds two decentering terms to the original model. The lemmas also estimate the upper bounds for the introduced error due to that shift. For a reasonable shift magnitude, this error remains within subpixel accuracy. For example, for a \(512 \times 512\) image with a big distorting value of \(k_1 = 1 \times 10^{-5}\), Lemma 1 states that even for a value of \(\Delta = 8\) the introduced error is still less than 1 pixel throughout the image.

\(^3\)The more familiar Euclidean norm could be used as well, but the former is more convenient for our work.
These results pave the way to the following proposition.

**Proposition 1:** The two decentering distortion coefficients \( p_1 \) and \( p_2 \) can compensate for the uncertainty in localizing the lens distortion center.

The proof of this proposition is straightforward from the above lemmas. A deviation in the location of the distortion center has no significant effect on the decentering distortion component of the model (Lemma 2) and its effect on the radial component is compensated for by \( p_1 \) and \( p_2 \) (Lemma 1). The maximum amount of the allowable uncertainty can be roughly estimated from the two lemmas by requiring the upper bounds of the approximation errors be within the required accuracy (say, subpixel accuracy).

Note that Lemma 1 conveys conclusion similar to Stein’s work [26] on the effect of allowing the distortion center to be different from the principal point. However, it more generally addresses the effect of setting the distortion center to a location different from its true location under radial distortion. Furthermore, Lemma 2 addresses the same effect but under lens decentering distortion, which was not studied in [26]. Here, we also, unlike [26], provide quantitative assessment of the error introduced.

We have done some simulations to validate this result. A set of five synthetic straight lines are generated across and near the boundaries of a 512 × 512 image. The lines were distorted by a strong distortion model \( U_1 \) consisting of \( K_3 = 2 \times 10^{-5} \) and \( p_1 = -3 \times 10^{-7} \). A new set of distorted lines were obtained from the original undistorted lines under a new model \( U_2 \) by shifting the distortion center by \( \delta \) pixels in both coordinates and by using a radial distortion coefficient of \( K_3 \) and two decentering coefficients of \( p_1 = K_3 \delta \) and \( K_3 \delta \). The rms error between the points of the two distorted sets was computed as a function of the shift \( \delta \) and is plotted in Fig. 4. Moreover, the closeness between the two models \( C(U_1, U_2) \) and the combined upper bound for the error between the two models as given by the two lemmas (the sum of the two bounds of the two lemmas) are plotted on the same graph. Clearly, even at shift of ten pixels in both coordinates on either side, the rms error is below 0.8 pixels. That is, across a 20 × 20 uncertainty region centered at the true distortion center, the second distortion model is still roughly within subpixel accuracy from the original model. The same conclusion is drawn considering the closeness measure between the two models. Note that the estimated upper bound for the approximation error plotted on the same graph is rather a loose bound in this experiment.

It was observed in our experiments, as well as in [4] and [8], that including both the distortion center and the decentering coefficients in the nonlinear search may lead to instability. Neither paper gave any explanation for this observation, while it was recommended in [8] that the estimation of the distortion coefficients be nested within a coarse-to-fine search for the distortion center in order to avoid the instability. In light of the last proposition, we are able to explain this observation as both the distortion center and the decentering coefficients tend to adjust for each other during the nonlinear search. Moreover, the proposition suggests a way to get around this situation by fixing the distortion center at an appropriate location (e.g., image center) and then using the two decentering distortion coefficients \( p_1 \) and \( p_2 \) to compensate for reasonable deviations of the center from the true location. As such, one can exclude the distortion center from the set of the distortion variables to be searched. This reduces the dimension of the search space without significant loss in accuracy, which leads to faster calibration.

As this result makes it even more straightforward to use the closed-form solution, we have derived in the last subsection, it would benefit all existing distortion calibration techniques [4], [6]–[8]. It is also interesting to note that on calibrating \( p_1 \) and \( p_2 \) of the model, typically one expects to get \( |p_1| > |p_2| \), which, according to Lemma 1, might not hold if the \( y \) deviation of the distortion center is bigger than that of the \( x \) deviation.

### C. Calibration Algorithm

To put all the pieces together, the whole calibration procedure is outlined in Algorithm 1.

**Algorithm 1** Outline of the calibration algorithm.

Input: a distorted image and order of distortion model \( p \).

1) Do subpixel edge detection to generate chains of edge points.
2) For each chain, compute the slope of the best-fit line.
3) Set the distortion center to a reasonable location (e.g., image center).
4) Form the set of equations \( Ax = b \) as described in Section IV-A (c.f., (8)). Use singular value decomposition to obtain the vector \( x \). The result is an initial estimate of the distortion coefficients.
5) Refine the distortion coefficients by applying a nonlinear optimization technique such as the LM al-

---

**Fig. 4.** RMS error in pixels between the line points, the closeness, and the upper bound of approximation error between the two models versus center shift in pixel.
algorithm to the distortion measure \( \xi \) (or \( \xi_g \) that to be described in the next section).

The input to the algorithm is a distorted image. However, to improve the calibration result, one can use several images acquired by the same lens from a variety of views. The extracted edge chains from all images, in this case, are then fed to the same algorithm outlined above. The order of the model is selected experimentally. Moreover, to refine the distortion coefficients obtained from the closed-form solution, our calibration algorithm uses \( \xi \) (7) or \( \xi_g \) (13). However, \( \xi \) (2) or similar objective functions can be used as well with comparable results.

V. CALIBRATION FROM IMAGE GRADIENTS

In this section, we follow the methodology of Section III to devise a calibration method using the distorted image gradients. We exploit a typical assumption that the image intensity gradient is parallel to the edge normal. As such, the image gradient at the curve points conveys some useful information on the straightness of the curve. Let \( I(x, y) \) and \( \hat{I}(x, y) \) be the given distorted image and the unknown undistorted image, respectively. Let \( (x^u_i, y^u_i) \) and \( (x^v_i, y^v_i) \) be points in \( I \) and \( \hat{I} \), respectively, that are related according to (1). Since the considered geometric lens distortion model would affect the location of the point but not its intensity, we have \( I(x^d_i, y^d_i) = \hat{I}(x^u_i, y^u_i). \) Thus

\[
\frac{\partial I(x^d_i, y^d_i)}{\partial x^d_i} = \frac{\partial \hat{I}(x^u_i, y^u_i)}{\partial x^u_i} \frac{\partial x^u_i}{\partial x^d_i} + \frac{\partial \hat{I}(x^u_i, y^u_i)}{\partial y^u_i} \frac{\partial y^u_i}{\partial x^d_i},
\]

(10)

That is, \( I_{x^d} = \hat{I}_{x^u} \frac{\partial x^u}{\partial x^d} + \hat{I}_{y^u} \frac{\partial y^u}{\partial x^d} \), where \( I_{x^d} \) \((\hat{I}_{x^u})\) is the \( x \) component of the gradient of the distorted (undistorted) image \( I(x^d_i, y^d_i) \) \((\hat{I}(x^u_i, y^u_i))\) with the indices dropped for simplicity. Similarly, \( I_{y^d} = \hat{I}_{x^u} \frac{\partial x^u}{\partial y^d} + \hat{I}_{y^u} \frac{\partial y^u}{\partial y^d} \). Solving for the \( x \) and \( y \) components of the gradient of the undistorted image in terms of those of the measurable distorted image yields

\[
\hat{I}_{x^u} = I_{x^d} \frac{\partial x^u}{\partial x^d} - I_{y^d} \frac{\partial x^u}{\partial y^d}, \quad \hat{I}_{y^u} = I_{y^d} \frac{\partial x^u}{\partial x^d} - I_{x^d} \frac{\partial y^u}{\partial y^d}.
\]

(11)

Thus, the gradient orientation \( \hat{g}(x^d_i, y^d_i) \) in the undistorted image, denoted by \( \hat{g} \), at the corresponding point \((x^d_i, y^d_i)\) is found to be a function of the gradient orientation of the distorted image, denoted by \( g \), at the corresponding point \((x^d_i, y^d_i)\)

\[
\hat{g}(x^d_i, y^d_i) = g(x^d_i, y^d_i) \frac{\partial x^u}{\partial y^u} - g(x^d_i, y^d_i) \frac{\partial y^u}{\partial y^u}.
\]

(12)

Therefore, if \((x^d_i, y^d_i)\) and \((x^d_j, y^d_j)\) are two edge points on the same curve in the distorted image, straightening the curve in the ideal image should make \( \hat{g}(x^d_i, y^d_i) = \hat{g}(x^d_j, y^d_j) \). The undistorted gradient orientations at the latter points are obtained from (12), since all the quantities involved can be computed from the distorted image or the assumed distortion model. Given a curve of edge points \((x^d_i, y^d_i); i = 1, \ldots, N\) that should belong to the same line in the undistorted image, the gradient-based distortion error per curve that needs to be minimized to search for the distortion parameters is

\[
\xi_g = \sum_{i=2}^{N} \left( \hat{g}(x^d_i, y^d_i) - \hat{g}(x^d_{i-1}, y^d_{i-1}) \right)^2.
\]

(13)

This distortion measure can be minimized using the techniques described in Section IV over all extracted image curves; closed-form solutions can be derived as before, then refined by nonlinear optimization algorithms to improve the accuracy.

Intuition suggests that the distortion criterion \( \xi_g \) must be related to \( \xi \) that we derived in (7). The answer to this is an emphatic yes. While the error criterion \( \xi_g \) forces the normals to the imaged line at its different points to be parallel, \( \xi_g \) obliges the line tangents at the same points to be in the same direction. The end result will be the same, a better “straightened” line. More explicitly, the term \( \frac{\partial g(x^d_i, y^d_i)}{\partial x^d_i} \) in (6) represents the slope to the tangent to a distorted curve in the image, whereas its counterpart, \( g(x^d_i, y^d_i) = \hat{g}(x^d_i, y^d_i) \), is simply the direction of the normal to the curve tangent. As such, \( \frac{\partial g(x^d_i, y^d_i)}{\partial x^d_i} \hat{g}(x^d_i, y^d_i) = -1 \). Similarly, we have \( \hat{g}(x^d_i, y^d_i) \hat{g}(x^d_i, y^d_i) = -1 \) at the corrected image point corresponding to \((x^d_i, y^d_i)\). Accordingly

\[
\hat{g}(x^d_i, y^d_i) = \frac{-1}{\hat{g}(x^d_i, y^d_i)} = -\frac{\partial x^u}{\partial y^u} + \frac{\partial y^u}{\partial x^u}, \quad \frac{\partial x^u}{\partial y^u} - \frac{\partial y^u}{\partial x^u}.
\]

which boils down after simple algebra to (12).

VI. ROBUST CALIBRATION

The distortion calibration method requires a number of chains of edge points that correspond to real 3-D linear segments. To meet this requirement, some user involvement in one form or another is needed. The user ought to select the edge chains that are projections of straight lines in the scene [5], [8]. Moreover, some “sample” points from each selected chain may be picked out and fed to the calibration algorithm [8]. Besides, a number of system parameters, such as edge linking thresholds, may need manual tuning.

In absence of any user involvement, a fully automatic method should be more tolerant, in other words, robust, to erroneous data that might enter the estimation algorithm in different forms:

- image curves that are mistakenly considered as projections of 3-D linear segments;
- image curves that do really correspond to 3-D linear segments but are linked together as one chain after the edge linking process. Using a smaller threshold in the edge linking and polygonal approximation process can do reduce this possibility but would yield smaller segments that may contain more noise than useful information about distortion. With a robust estimation method, one can risk using a bigger linking threshold to produce longer imaged segments that contain more useful information.

\(^4\text{Again, here, a more general form would be } \sum_{i=1}^{N} \sum_{j=1}^{N} (\hat{g}(x^d_i, y^d_i) - \hat{g}(x^d_j, y^d_j))^2.\)
Fig. 5 shows some examples of these erroneous data found in the extracted chains from the image. These errors will result in the presence of outliers, which will severely degrade the distortion estimation algorithm if one directly applies the methods described above or in the literature since they are all least-squares techniques. We are aware of only one work [6] that tried to reduce the effect of outliers on distortion calibration. In that work, Devernay and Faugeras used a smaller polygonal approximation threshold to produce the edge chains that are to be used by the optimization process. Then, by repeating distortion minimization and polygonal approximation on the undistorted edges using the resultant distortion parameters, many outliers can be eliminated and longer, more useful segments can be obtained and, thus, more accurate calibration. Their technique relies on that undistorting the edges after the optimization would make identifying outliers during the next polygonal approximation easier. However, this would not be the case when the image has severe distortion, when many 3-D segments are broken into smaller edges, or when too many outliers are found in the data. In any of these cases, the distortion parameters obtained after the first iteration will be highly perturbed, and will not make the identification of outlying points in the beginning of the next iteration any easier.

In this section, we propose an automatic method for lens distortion calibration based on robust estimators. These estimators started in the field of mathematical statistics [16], [27], [28] and have gain popularity over the past decade in the computer vision community because algorithms data are unavoidably error prone [29], [30]. The interested reader is referred to [31] for a comprehensive review.

A. Robust Method

The LMedS method estimates the parameters by solving the nonlinear minimization problem: \( \min \ \text{median}_i r_i^2 \), where \( r_i \) denotes the residual of the \( i \)th datum. As the LMedS method theoretically has the largest possible breakdown point (0.5) [16], we base our proposed method on it. The algorithm which we have implemented generally follows the structure outlined in [16] and is summarized below.

Given \( q \) edge points, a Monte Carlo-type technique is used to draw \( m \) random subsamples of \( q \) different points. For each subsample, indexed by \( J \), we determine the distortion parameters \( \mathcal{P}_J \) using the methods described before. For each \( \mathcal{P}_J \), we can determine the median of the squared residuals, denoted by \( M_J \) with respect to the whole set of points. We retain the estimate \( \mathcal{P}_J \) for which \( M_J \) is minimal among all \( m \) \( M_J \)s. The number of subsamples \( m \) should be big enough such that at least one of the subsamples is “good.” A subsample is good if it consists of \( q \) good edge points. Assuming that the whole set of points may contain up to a fraction \( \epsilon \) of outliers, the probability that at least one of the \( m \) subsamples is good is given by

\[
P = 1 - [1 - (1 - \epsilon)^m]^m.
\]

By requiring that \( P \) must be near 1 (say, 0.99), one can determine \( m \) for given values of \( \epsilon \) and \( m \)

\[
m = \frac{\log(1 - P)}{\log[1 - (1 - \epsilon)^m]}.
\]

The LMedS efficiency is poor in the presence of Gaussian noise [16]. To compensate for this deficiency, one first make a good, robust estimate of the standard deviation of the errors of good data (inliers). This estimate is related to the median of the absolute values of the residuals, given by [16]:

\[
\hat{\sigma} = 1.4826[1 + 5/(n-q)]\sqrt{M}\,,
\]

where \( M \) is the minimal median. Any data item whose error is larger than a certain number (e.g., 2.5–3.0) of \( \hat{\sigma} \) can be considered as an outlier and removed. The distortion parameters are finally estimated by applying the distortion calibration algorithm once again on the inlying points.

B. Implementation Details

Here, we discuss in more detail some issues related to the implementation of the robust algorithm. The first issue is how to compute the residual \( r_i \) for each point. The residual of a particular point should reflect its own contribution to the fitting error of the model. The previously proposed distortion measures, e.g., \( \xi_p \) in (2), and the proposed measures \( \xi_s \) or \( \xi_d \) gauge the distortion carried by a point but in accordance with one or more points on the same imaged line. For example, if the best-fit line parameters of a curve are estimated from some points, one of which is outlying, the distortion error of any point on the curve, even good ones, will be skewed. Therefore, one cannot compute the residual \( r_i \) directly from (2), (7) or (13). Instead, once we solve for the distortion parameters for a subsample using any of our methods, the residual \( r_i \) for each point is computed as the distance from the point to the line robustly estimated from the curve points. That is

\[
r_i^2 = (x_i^2 \sin(\theta_i) + y_i^2 \cos(\theta_i) - \hat{\rho_i})^2
\]

where \( \theta_i \) and \( \hat{\rho_i} \) are the robustly estimated best-fit line parameters of the curve 1 after undistorting the curve points by the parameters obtained from the subsample. The best-fit parameters \( (\theta_i, \hat{\rho_i}) \) are computed using the LMedS estimator following the outline of the previous subsection. The robustly estimated line slope is also used in the linear approach of Section IV-A.
to find initial solution for each subsample before the nonlinear optimization step.

As said previously, computational efficiency of the LMedS method can be achieved by applying a Monte Carlo-type technique. However, the points of a subsample have to be selected properly to avoid wasting time to evaluate an improper subsample, e.g., one having each point from a different imaged line. Also, the situation of having the points very close to each other should be avoided as the distortion parameters from such points is highly instable and the result is useless. To select the points of a subsample, one should, thus, take care of two concerns. On one hand, the points of a subsample ought to be distributed across the image in order that the obtained parameters be not biased by the region from which the points come. On the other hand, the points selected should provide enough constraints to solve for the distortion parameters. For example, we need at least two points from any line to impose one constraint on the distortion parameters; see (7).

In order to consider these concerns and achieve higher efficiency and stability, we used a regularly random selection method based on bucketing techniques [32], which works as follows. The minimum and the maximum of the coordinates of the extracted edge points in the image are calculated. Then the region within these limits is divided into $b \times b$ buckets (in our implementation, $b = 2$). Each extracted edge chain is attached to the bucket that includes most of the chain points. Buckets having no points are excluded. To generate a subsample of $q$ points, we randomly select $q/2$ buckets, and then randomly choose a chain from each selected bucket. From each chosen chain, two points are picked out at random, one from the first half of the chain and another from the second half. Thus, we end up with $q$ selected points per sample.

Assuming that the bad points are uniformly distributed in the image, the required number of subsamples $m$ can be computed from (15). However, the number of edge points in one bucket may be quite different from that in another. As a result, a point belonging to a bucket having fewer points has a higher probability to be selected. It is, thus, preferred that a bucket having many points has a higher probability to be selected than a bucket having few points, in order that each point has almost the same probability to be selected and, thus, formula (14) holds. To realize this, we divide the interval $[0 1]$ into $q/2$ intervals such that the width of the $i$th interval is equal to $n_i/\sum n_i$, where $n_i$ is the number of points attached in the $i$th bucket. During the bucket selection procedure, a number produced by a $[0 1]$ uniform random generator falling in the $i$th interval implies that the $i$th bucket is selected. Once a bucket has been selected, a similar procedure is used to select a chain from that bucket.

One question remains: How many subsamples are required? To draw enough points for a subsample to solve for $p$ distortion parameters, we need at least $p$ constraints on the distortion parameters. So need at least $2p$ points with each pair laying on the same imaged line. Moreover, we find that using a little bit more points per subsample improves the results, so in general, we take the number of points per subsample $q \geq 2p$. For example, we take $q = 8$ if $p = 2$ or $3$. In this case, if we assume $\epsilon \geq 30\%-40\%$ and require $P = 0.99$, thus, from (15), $m = 78–272$.

Finally, an outline of the robust calibration algorithm is given as Algorithm 2.

VII. EXPERIMENTAL RESULTS

In this section, the performance of our technique is assessed using both synthetic and real image data.

**Algorithm 2 Outline of the robust calibration algorithm.**

**Input:** a distorted image and order of distortion model $p$.

1) Do subpixel edge detection to generate chains of edge points.
2) For each chain, compute the line best-fit parameters using a typical LMedS estimation procedure [16].
3) Set the distortion center to a reasonable location (e.g., image center).
4) Compute the required number of subsamples from (15).
5) **for** each subsample do
   a) Select the points of the subsample using the bucketing technique described before.
   b) Estimate the distortion coefficients as outlined in Algorithm 1.
   c) Compute the median of the squared residuals for all points at the estimated coefficients using (16).
6) **end for**
7) Set $M^*$ to the minimal median over all subsamples.
8) Compute $\hat{\sigma}$ and identify outliers.
9) Recompute the distortion coefficients by applying Algorithm 1 to all data points after discarding the identified outliers.

The synthetic images provide exact knowledge of line positions, orientations, and distortion parameters, so precise quantitative evaluation of performance is possible. The performance on real images is shown to demonstrate the practical implementation of the technique.

A. Synthetic Data

A $320 \times 242$ image consisting of ten lines is used as a test image. The lines were generated with random orientations and positions. Using known distortion parameters, points (about 880 in total) sampled from the lines were distorted; see Fig. 6(a). To simulate errors in feature extraction, the location of each point was then perturbed in a random direction by a distance
governed by a Gaussian distribution with zero-mean and standard deviation, $\sigma$, expressed in pixels. Such a distribution results in an expected value for the perturbation distance of approximately $0.8\sigma$. As a matter of interest, in the many tests carried out here, the highest perturbation distance was found to be $5.1\sigma$.

We then used our linear approach outlined in Section IV-A to estimate the distortion parameters from the noisy data. We also used the LM algorithm to minimize the distortion measure in (7) starting from a reasonable guess of the distortion parameters (as described in Section IV). The estimated distortion parameters were then used to undistort the noiseless undistorted image, see the result in Fig. 6(b). Table I shows the true distortion parameters used to create the image and the results obtained at $\sigma = 0.1$ pixels. The second row shows the results of the linear approach, followed in the third row by the nonlinear optimization technique. For the linear approach, the distortion center was taken at the true location. The run time in seconds on an SGI-O2 machine for the linear approach is 0.05, while it is 3.2 s for the nonlinear approach. Clearly, the linear solution provides a faster response but the nonlinear optimization algorithm provides more accurate results.

Comparing the ground truth distortion parameters and the estimated ones does not give an intuitive feel of how much distortion remain in the image. In addition, different sets of the distortion parameters may give rise to virtually similar undistorted images. Therefore, we have considered the rms error in pixels between the true and the undistorted line points as an accuracy measure [5], [8]. We used this measure, denoted $\varepsilon$, to evaluate the performance of our proposed approaches versus different noise levels. The experiments were conducted with $\sigma$ varying from 0.0 to 3.0 in steps of 0.5 pixels. For the linear approach, we fixed the distortion center at the image center (different from the true location shown in the first row of Table I) and solved for one radial and two decentering distortion coefficients. The obtained solution were further refined by minimizing the distortion measure in (7) using the LM algorithm. We compared the results of the two methods with those of the LM algorithm minimizing two previously proposed objective functions: the sum of square distances of the points from their best-fit lines [4], [7] and the $\chi^2$ of the least square approximation used in [6] ($\mathcal{L}_\chi$). For the latter two objective functions, the LM algorithm solved for distortion model consisting of ($c_x$, $c_y$, $\kappa_1$, and $\kappa_2$). For each value of $\sigma$, the calibration was carried out 20 times, each with a different seed number for the random number generator. The average of the accuracy measure $\varepsilon$ for the four methods computed over the 20 trials is plotted in Fig. 7. The two approaches minimizing $\mathcal{L}_\varepsilon$ and $\mathcal{L}_\chi$ showed similar performance against noise since both are using two different versions of the same error function. However, both performed much worse than our linear and nonlinear approaches. The main reason is that the two approaches included the distortion center in the search space, which led to bigger errors at high noise levels. On the other hand, in our approach, the distortion center is kept fixed while two decentering distortion coefficients are used to make up for its deviation from...
its true location. In addition, although the model that our non-linear technique solved for was not exactly the model used to distort the image, the calibrated model was very comparable in performance to the true model ($\varepsilon$ is virtually zero at low noise levels), and at the same time, we gain more stability especially at higher levels of noise. It is also clear from Fig. 7 that the non-linear optimization stage has indeed improved on the outcome of our linear approach.

The same series of experiments were carried out for the three objective functions, $\xi_L$, $\xi_c$, and the proposed one $\xi_c$, by applying the LM optimization algorithm to the objective functions, all of them starting from the solution of our proposed linear approach. For the three approaches, the distortion model was the same, which consisted of $\kappa_1$, $p_1$, and $p_2$, while the distortion center was fixed at the image center. This was also repeated for different positions of the true distortion center (to test the effect of center shift on the performance). Fig. 8 shows the average performance of the three objective functions against noise level at four center positions. Also shown on the same graph is the performance of each objective function after initiating the non-linear minimization at the initial guess described in Section IV (almost distortion-free model).

Clearly, the figure shows that the performance of the two objective functions $\xi_L$ and $\xi_c$ has considerably improved as opposed to the previous one in Fig. 7. This may be attributed to fixing the distortion center and allowing the two decentering coefficients to counterbalance its localization error. Here, again, the calibrated model produced by the three objective functions, although different from the true distorting model, has shown very comparable performance to the original model (as evident from the low $\varepsilon$ error at low noise for the different center positions) with much better noise stability.

Comparing the results of minimizing each objective function starting from the closed-form solution and starting from...
the almost-distortion-free model shows very comparable performances within 0.2–1.0 pixels in favor of the closed-form initialization (rather not visually notable on the graph because of the graph scale). This can be explained as that the linear approach moved the estimation algorithm faster near the correct solution. The end result, however, was very similar to the slower optimization routine starting from the farther starting point of the almost-distortion-free model (the farther the starting point, the more probable the nonlinear optimization routine gets stuck in a local minimum away from the correct solution, which luckily did not happen here). This is also supported by comparing the measured time performance in both cases, which reveals an average speedup provided by the closed-form initialization in the order of 3–4 times.

One more thing to read from Fig. 8 is the slightly better performance of both the $\xi_f$ and $\xi_c$ functions in terms of $\varepsilon$ over the proposed $\xi_s$ measure when the search space for all of them was the same. This can be attributed to the fact that $\xi_s$ mainly emphasizes the slope information of the imaged lines but not the intercept information, while the $\varepsilon$ criterion weighs the two components, slope and intercept, both of which are exploited in $\xi_f$ and $\xi_c$. This comes as the price one pays on using $\xi_s$ in return for more reliability of the slope information across wide range of distortion, c.f., Figs. 2 and 3. Nevertheless, given that the edge points can be routinely located by automated techniques with an feature extraction accuracy of better than $\sigma \approx 0.5$, the $\varepsilon$ criterion for the three objective functions will be nearly the same.

B. Real Images

Our approach is also applied to several real images captured by an IndyCam in our lab. The acquired images are 640 × 480 pixels and typically have noticeable lens distortion due to the cheap wide-angle lens. Since image distortion is sometimes less than a pixel, we used an edge detection method with a subpixel accuracy [33], which is quite robust to noise (with reported theoretical accuracy of 0.1 pixel [33]), followed by edge linking process. We used relatively a big tolerance for polygonal approximation; the maximum distance between edge points and the segments joining both ends of the edge must be less than 3–5 pixels. This would result in longer edge chains with possibly joined segments to be taken care of by the robust estimation algorithm. We also used a threshold of about 70 on the length of the resulting edge chains for the 640 × 480 image because small segments may contain more noise than useful information about distortion. Moreover, because of the corner rounding effect [6] due to edge detection, about 5–8 edge points at both ends of each chain are thrown away. Fig. 9 shows a distorted image and the undistorted image using the robust distortion algorithm using the measure $\xi_s$. Some straight lines are imposed in dashed-lines on the image to help demonstrate the effect of distortion and correction on the image. As can be seen in the undistorted image, straight lines in the scene now map to straight lines in the image. Another example is shown in Fig. 10. The image was successfully undistorted using the robust approach utilizing the objective function $\xi_f$ where the image gradients were computed using a recursive Gaussian filter [34].

The approach has also been tested with several images with severe lens distortion. We used the robust approach with $\xi_f$ to undistort the image shown in Fig. 11(a). The extracted edge chains used in the estimation are shown in Fig. 11(b). It is interesting to see the performance of only the linear distortion on this highly distorted image. The result of the robust algorithm using only the closed-form solution without any nonlinear optimization is shown in Fig. 11(c). Surprisingly, the closed-form solution provided a fairly good undistorted image. Afterwards, the nonlinear minimization of $\xi_f$ gave the image a final polish as seen in Fig. 11(d). Another example is shown in Fig. 12. This image was undistorted using the robust estimation algorithm with the objective function $\xi_c$. Clearly, the robust algorithm alleviated the influence of some outliers in the automatically extracted data as shown in Fig. 12(a). In Table II, we summarize the experiments on the above real images. We give the number of segments detected and used for the robust calibration algorithm,
the number of edge points (edgels) forming these segments and the CPU time of both the distortion calibration and image undistortion in minutes on a SGI-O2 workstation. The time shown is merely indicative, as we have not tried to optimize the code.
TABLE II
SUMMARY OF EXPERIMENTS ON DIFFERENT SCENES: NUMBER OF INPUT SEGMENTS AND EDGELS, AND TIME (IN THE FORMAT MINUTE : SECOND) NEEDED FOR CALIBRATION AND UNDISTORTION

<table>
<thead>
<tr>
<th>Scene</th>
<th>Segments</th>
<th>Edgels</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>desk</td>
<td>30</td>
<td>2234</td>
<td>5 : 13</td>
</tr>
<tr>
<td>mouse</td>
<td>34</td>
<td>3006</td>
<td>7 : 13</td>
</tr>
<tr>
<td>exhibit</td>
<td>45</td>
<td>746</td>
<td>1 : 18</td>
</tr>
<tr>
<td>hallway</td>
<td>33</td>
<td>2330</td>
<td>5 : 56</td>
</tr>
</tbody>
</table>

This time of calibration depends on the number of edge points used and the number of distortion coefficients to be calibrated, whereas the time of image undistortion depends essentially on the image size. The calibration algorithm can be speeded up considerably by means of parallel computing, because the processing for each subsample can be done independently. However, it is still considerably less than the time taken by some other techniques. For example, Swaminathan and Nayar [8] who used a coarse-to-fine exhaustive search for the distortion center to avoid the instability of its search along with the other coefficients reported a run-time of about 20 min to calibrate an image similar to that of Fig. 12. This time does not include the manual line and point selection and other data preparation for the estimation algorithm, which is totally not needed in our approach thanks to the robust LMedS estimation method.

VIII. CONCLUSION

In this paper, we have addressed the problem of nonmetric calibration of lens distortion with some new results in the theory and computation aspects of the problem. In that regard, the paper makes the following contributions.

- We have derived two new distortion measures that can be used to calibrate the distortion parameters of the lens; one is based on the slopes of the imaged lines in the distorted image and the other is based on the image gradients. The first is used to constrain the imaged line’s tangents at its different points to be in the same direction, while the other is to force the normals to the imaged line points to be parallel. The end result will be the same, a better “straightened” line and undistorted image. From these two measures, we have shown how fast, analytic solutions for the distortion coefficients can be obtained. This represents a major advantage of our approach over the other existing nonmetric calibration techniques. Our experimental results on synthetic and real data show that the analytic solution can provide good estimate of the distortion coefficients. This solution can also serve as an initial point for a following refining stage based on nonlinear optimization of our proposed distortion measures or even previously used error measures.

- It is shown that the deviation of the distortion center from its true location under both lens radial and decentering distortion is equivalent to adding two additional decentering distortion terms. This explains, for the first time, the instability reported in previous efforts [4], [8] when including both the distortion center and the decentering coefficients in the nonlinear optimization. Moreover, this finding suggests a way to get around this situation by fixing the distortion center at an appropriate location (e.g., image center) and then using the two decentering distortion coefficients to compensate for reasonable deviations of the center from the true location. This reduces the search space of the calibration problem without sacrificing the accuracy and produces more stable and noise-robust results as was verified by the experimental results.

- A robust approach to distortion calibration is also proposed. The approach exploits a very robust technique—the LMedS—to discard outliers in the data that might enter the estimation algorithm in different forms. As such, the proposed approach is able to proceed in a fully automatic manner, whereas almost all existing nonmetric distortion calibration methods need some user involvement for data preparation in one form or another.
Currently, the distortion model order is selected experimentally based on inspection of the input and resulting images. Our current research direction is directed to identifying the best distortion model order based on a statistical inference criterion such as geometric AIC [35]. The advantage of this criterion is that it can estimate the most probable distortion model taking into account the complexity of the model along with the residual of data fitting to the model. Our early results on this are published in [36].

**APPENDIX**

**PROOF OF LEMMAS 1 AND 2**

In this appendix, we provide proof of the two Lemmas of Section IV-B.

**Lemma 1:**

*Proof:* Assume radial distortion model with dominant $k_3$.

From Definition 1, we can, thus, drop all distortion coefficients other than $k_3$. Then

\[ x'' = x''d + (x''d - c_x)k_3 \left[ (x''d - c_x)^2 + (y''d - c_y)^2 \right]. \]

Let $(c_x', c_y') = (c_x - \Delta c_x, c_y - \Delta c_y)$ be the new shifted distortion center. Thus

\[ x'' = x''d + (x''d - c_x' - \Delta c_x)k_3 \left[ (x''d - c_x')^2 + (y''d - c_y')^2 \right]. \]

i.e.

\[
x'' = x''d + (x''d - c_x')k_1 f_4d + (-\Delta x k_1) \left[ f_4d^2 + 2(x''d - c_x')^2 \right] \\
+ 2(-\Delta y k_1)(x''d - c_x')(y''d - c_y) \\
+ (x''d - c_x')(k_3 \Delta^2 + 2k_1 \Delta_1^2) \\
+ 2\Delta_x \Delta_y k_1 (y''d - c_y) - k_1 \Delta x \Delta^2 
\]

where $f_4d = (x''d - c_x')^2 + (y''d - c_y')^2$ and $\Delta^2 = \Delta x^2 + \Delta y^2$.

Analogously

\[
y'' = y''d + (y''d - c_y')k_1 f_4d + (-\Delta y k_1) \left[ f_4d^2 + 2(y''d - c_y')^2 \right] \\
+ 2(-\Delta x k_1)(x''d - c_x')(y''d - c_y) \\
+ (y''d - c_y')(k_3 \Delta^2 + 2k_1 \Delta_1^2) \\
+ 2\Delta_x \Delta_y k_1 (x''d - c_x') - k_1 \Delta y \Delta^2. 
\]

The second term of (17) and (18) is the radial distortion term with respect to the new center $(c_x', c_y')$. The third and fourth terms are decentering distortion terms with $p_1 = -\Delta x k_1$ and $p_2 = -\Delta y k_1$. Let $U$ denote the original model and let $U'$ be the new distortion model resulting from the center shift and consisting of the first three terms in the two equations. Clearly, the fifth and following terms in the two equations contribute to the deviation between the models.

Note, however, that in (17) and (18), the fifth and following terms are considerably smaller than the third and fourth terms since the third and fourth terms are quadratic in the distance of a point from the distortion center and only linear in the center shift. In contrast, the following terms are at most linear in the point distance and at least quadratic in the shift. Considering that distortion is strongest far away from the distortion center, at such places the distance of points to the center is much larger than the center shift.

The approximation error between $U$ and $U'$ is given from (9) as $e = \max(e_x, e_y)$, where

\[
e_x = \max_{x'', y''d} \left( (x''d - c_x') \left( k_3 \Delta^2 + 2k_1 \Delta_1^2 \right) \\
+ 2\Delta_x \Delta_y k_1 (y''d - c_y) - k_1 \Delta x \Delta^2 \right), \text{ and} \\
e_y = \max_{x'', y''d} \left( (y''d - c_y') \left( k_3 \Delta^2 + 2k_1 \Delta_1^2 \right) \\
+ 2\Delta_x \Delta_y k_1 (x''d - c_x) - k_1 \Delta y \Delta^2 \right). 
\]

An upper bound for the approximation error can be computed as follows. Let $\Delta = \max(\Delta x, \Delta y)$. Since $(c_x', c_y')$ is often set to $(W/2, H/2)$. Then

\[
e_x \leq \max_{x'', y''d} \left( |x''d - c_x'| \|k_1\| |2\Delta^2 + 2\Delta_1^2| \\
+ 2\|k_1\| |y''d - c_y| + 2\|k_1\| \Delta^3 \right). \\
e_y \leq \|k_1\| \left( (W + H) \Delta^2 + 2\Delta^3 \right). 
\]

Since, typically, $H \leq W$, then the required upper bound for the approximation error is

\[
e = \max(e_x, e_y) \leq \|k_1\| \left( (W + H) \Delta^2 + 2\Delta^3 \right). 
\]

**Lemma 2:**

*Proof:* Similar to the above proof, from the original distortion model $U$ after the center shift

\[
x'' = x''d + p_1 \left[ f_2 + 2(x''d - c_y)^2 \right] + p_2 \left[ (x''d - c_x)(y''d - c_y) \right] - [6p_1 \Delta x + 2p_2 \Delta y] (x''d - c_x) \times \left( y''d - c_y \right) + \left( p_1 \Delta^2 + 2p_2 \Delta_1^2 \right) + 2p_2 \Delta_x \Delta_y. 
\]

Analogously

\[
y'' = y''d + p_2 \left[ f_2 + 2(y''d - c_y)^2 \right] + p_1 \left[ (x''d - c_x)(y''d - c_y) \right] - [2p_2 \Delta_x + 2p_1 \Delta y] (x''d - c_x) \times \left( y''d - c_y \right) + \left( p_2 \Delta^2 + 2p_1 \Delta_1^2 \right) + 2p_1 \Delta_x \Delta_y. 
\]

Let $U'$ be the distortion model consisting of the first three terms in (19) and (20). Obviously, it represents the decentering distortion model that we started with but with respect to the new center $(c_x', c_y')$. Note that the second and third terms are considerably bigger than the following terms since they are quadratic in the distance of a point from the distortion center, whereas the following terms are at most linear in the point distance and contribute to the error between the two models. The approximation error is computed again from (9) as $e = \max(e_x, e_y)$, for which an upper bound can be computed, as we did before

\[
e_x \leq \max_{x'', y''d} \left( 6p_1 + 2p_2 \|x''d - c_x| \Delta + 2p_1 + 2p_2 \|y''d - c_y| \Delta \\
+ \left[ p_2 + 2p_1 \|x''d + 2p_2 \|x''d \Delta \right] \Delta^2 \right). 
\]

Similarly

\[
e_y \leq \left( 2p_1 + 2p_2 \right) \Delta^2 + \left( (W + H)p_1 + (W + H)p_2 \right) \Delta. 
\]

Then, the upper bound is found from $e = \max(e_x, e_y)$ as stated in Lemma 2.