Centi-pixel accurate real-time inverse distortion correction

Jason P. de Villiers*[a,b], F. Wilhelm Leuschner[b], Ronelle Geldenhuys[b]

[a]Council for Scientific and Industrial Research, Meiring Naude St, Pretoria, South Africa;

[b]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa;

ABSTRACT

Inverse distortion is used to create an undistorted image from a distorted image. For each pixel in the undistorted image it is required to determine which pixel in the distorted image should be used. However the process of characterizing a lens using a model such as that of Brown, yields a non-invertible mapping from the distorted domain to the undistorted domain. There are three current approaches to solving this: an approximation of the inverse distortion is derived from a low-order version of Brown's model; an initial guess for the distorted position is iteratively refined until it yields the desired undistorted pixel position; or a look-up table is generated to store the mapping. Each approach requires one to sacrifice either accuracy, memory usage or processing time. This paper shows that it is possible to have real-time, low memory, accurate inverse distortion correction. A novel method based on the re-use of left-over distortion characterization data is combined with modern numerical optimization techniques to fit a high-order version of Brown's model to characterize the inverse distortion. Experimental results show that, for thirty-two 5mm lenses exhibiting extreme barrel distortion, inverse distortion can be improved 25 fold to 0.013 pixels RMS over the image.

Keywords: Inverse distortion, numerical optimization, distortion characterization, real-time

1. INTRODUCTION

Distortion is inherent in all images taken through lens systems. Many applications, such as machine vision, image stitching, product defect detection; motion capture, better video compression, 3D measurement/reconstruction, and digital archiving can all benefit from compensating for this distortion. The classical lens distortion model is that of Brown[1,2] and Conrady.[3] The equation for Brown's model is:

$$
\begin{aligned}
x_u =& x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \left(P_1(r^2 + 2(x_d - x_c)^2) + 2P_2(x_d - x_c)(y_d - y_c)\right)(1 + P_3 r^2 + \ldots) \\
y_u =& y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \left(2P_1(x_d - x_c)(y_d - y_c) + P_2(r^2 + 2(y_d - y_c)^2)\right)(1 + P_3 r^2 + \ldots))
\end{aligned}
\tag{1}
$$

where:

$$
\begin{aligned}
(x_u, y_u) &= \text{undistorted image point,} \\
(x_d, y_d) &= \text{distorted image point,} \\
(x_c, y_c) &= \text{centre of distortion,} \\
K_n &= \text{N}^{\text{th}} \text{ radial distortion coefficient,} \\
P_n &= \text{N}^{\text{th}} \text{ tangential distortion coefficient,} \\
r &= \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}, \text{ and} \\
\ldots &= \text{an infinite series.}
\end{aligned}
$$

[*] jdvilliers@csir.co.za; phone: +27 12 841 3738; www.csir.co.za

Optomechatronic Technologies 2008, Otani, Bellouard, Wen, Hodko, Katagiri, Kassegne, Kofman, Kaneko, Perez, Coquin, Kaynak, Cho, Fukuda, Yi, Janabi-Sharifi, Eds., Proc. of SPIE Vol. 7266, 726611 • © 2008 SPIE • CCC code: 0277-786X/08/$18 • doi: 10.1117/12.804771

This allows one to determine where any point in the distorted image would appear in the image plane if there was no lens distortion. However, in practice, one generates an undistorted image in the opposite manner. One starts with a blank slate for the undistorted image, and for every pixel determine which distorted pixel to use. In general it is unlikely that the calculated source distorted pixel has integer coordinates, and so some interpolation between the four adjacent pixels is necessary.

The ability to find the distorted pixel which corresponds to an undistorted picture is known variously as undistortion, distortion correction and inverse distortion. Inspection of Brown's distortion model in Eq (1) shows that except for trivial first order radial implementations with no tangential modelling the equation is nonlinear in terms of r^2. Thus although finding a distorted point's corresponding undistorted point is simple, the opposite is not true. Typically one guesses where the undistorted point is (possibly by calculating the correction of a distorted point at the undistorted point's position, and subtracting instead of adding that correction) and then adjusts that point until the distance between its corresponding undistorted point and the original undistorted point is acceptably close to zero. This involves an iterative numerical refinement of the guessed distorted position, which requires at least three distortion model calls per iteration to determine the error gradient. This is neither a quick nor a deterministic procedure, making it undesirable for real-time applications.

It is possible to use the above method to pre-generate a look up table. This entails a direct trade-off of accuracy versus memory usage, as a coarser sampling of image points in the look-up table will save memory usage at the expense of accuracy as errors are induced by interpolating between the values in the look up table.

Approximate inverse functions based on Taylor expansions of low-order versions of Brown's model have a simple function to generate the undistorted position from the distorted position and recquire only the storage of a few parameters. However this is at the expense of achieving accuracies of only 0.3 to 0.4 pixels.[4] Other approaches based on different distortion models such as Candocia's scale preserving model[5] have analytical inverse distortion solutions, although typically these are not suitable for real-time implementation. Candocia's model for instance requires solving two fifth order polynomials per undistorted pixel, which itself is an iterative numerical procedure.

This paper shows that it is possible to have a non-iterative, memory efficient, high accuracy, real-time inverse distortion correction. It based on the reuse of undistorted points which are generated in the lens distortion calibration process. High order Brown models are then used to characterize the mapping from the undistorted to distorted domains.

The rest of this paper is organised as follows: Section 2 describes the physical setup that was used to capture the data to verify the inverse distortion techniques. Section 3 provides a brief backgroud on the numerical optimization techniques used. Section 4 shows that high-order Brown models are both stable and provide better distortion characterization when proper numerical optimization methods are used. Section 5 details how these models and techniques can then be applied to model Inverse Distortion effectively. Finally, Section 6 places the results of this work in context.

2. EXPERIMENTAL SETUP

In order to verify the efficacy of high order models and modern optimization techniques, real world data was collected, and a metric to determine the level of distortion and undistortion formulated. These are presented below.

2.1 Cameras

A sample of 32 cameras each with a nominal focal length of 6mm and 76° by 56° field of view (FOV) frustum were used. The cameras have square pixels and the horizontal and vertical axes are orthogonal.

2.2 Optical reference jig

These cameras each captured the centroids of an optical reference jig which has 504 circular marks arranged in 6 concentric squares. Each camera captured 4 different views of the jig such that the entire field of view was covered. This is portrayed in Figure 1.

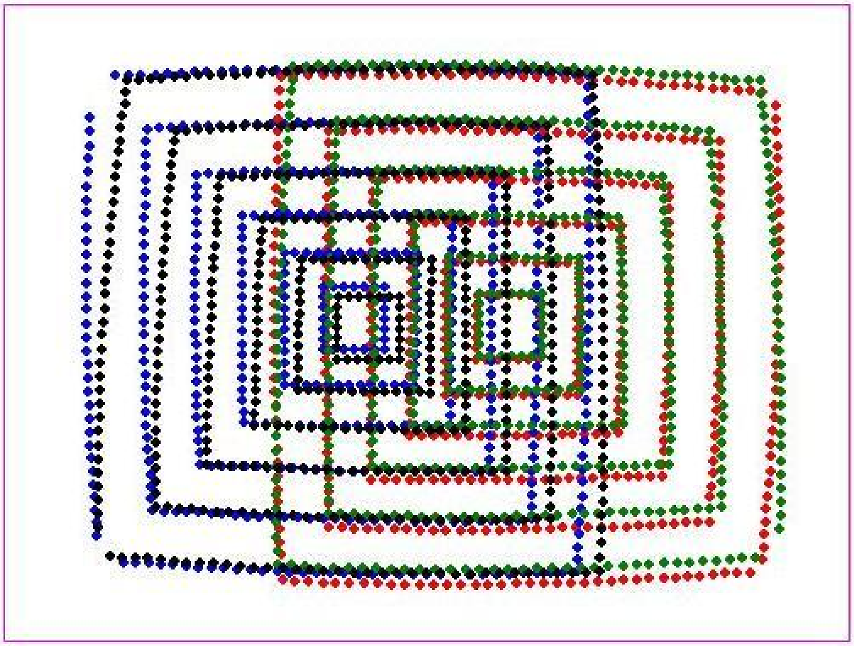

Figure 1. Depiction of captured image data, the outline represents the FOV of the camera

Figure 2 shows the distorted and undistorted points for one of the views of the optical reference jig. The undistorted positions were calculated using Brown's model with 3 radial terms and 3 tangential terms and optimal distortion centre. The difference in the top right corner is equal to 73 pixels, or 8.7% of the diagonal FOV.

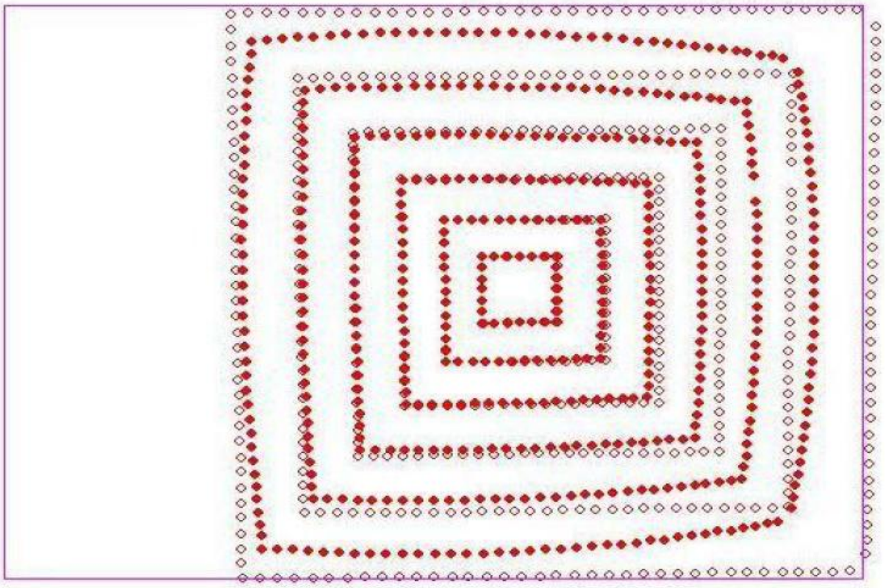

Figure 2. Comparison of distorted and undistorted points, the outline represents the FOV of the camera

2.3 Distortion measure

Brown proposed the plumb-line method in 1971[2] and it has been used almost universally since. Essentially, given points along a supposed-to-be-straight line, Brown fitted the least- squares straight line through the points, the error (for that line) was the sum of the perpendicular distances from the line. A slight adaptation to the above was used whereby the perpendicular distance between the point and the line is still used, but the line is expressed in the more tractable form of a unit direction vector and a point on the line. The following equation shows how the distance from the line is calculated.

$$dist^2 = (\|\bar{p_p} - \bar{p_l}\|_2)^2 - ((\bar{p_p} - \bar{p_l}) \cdot \bar{d_n})^2 \tag{2}$$

where

$dist =$ the distance from the line to the point in question,

$\bar{d_n} =$ unit direction vector of the line,

$\bar{p_l} =$ the point on the line, and

$\bar{p_p} =$ the point in question.

The Root Mean Square (RMS) of all the distances of the 2016 points off their corresponding lines is used as the distortion measure for that camera.

2.4 Undistortion measure

Given a set of distorted and undistorted points, a simple measure of how well the undistortion model works is the RMS of the distances between the actual distorted points and those the model generates from the undistorted points. Mathematically this is expressed as:

$$error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left([\| < x_{di}, y_{di} > -f(x_{ui}, y_{ui})\|_2]^2\right)} \tag{3}$$

where

$n =$ the number of points

$(x_{di}, y_{di}) = i^{th}$ distorted point,

$(x_{ui}, y_{ui}) = i^{th}$ undistorted point, and

$f() =$ function which generates distorted points from undistorted points.

3. NUMERICAL TECHNIQUES OVERVIEW

Equations (2) and (3) provide a means to quantify how much distortion a lens induces, and how effictive the undistortion is, respectively. Lower values are better, and this section describes the methods used to minimize these two metrics.

The Steepest Descent (SD) algorithm is a simple line search technique, where the search direction is the opposite of the gradient vector - the direction of steepest descent. The result is successive search directions being orthogonal and a theoretical infinite number of iterations required for convergence.

The Levenberg-Marqurardt (LM) algorithm[6,7] is a combination of SD and the classical Newtonian method for optimizing systems of equations, it combines the benefits of rapid convergence with SD's guaranteed eventual convergence. However it is known that in applications where the parameters are highly correlated, the algorithm performs poorly and may falsely claim to have converged due to the step size dropping below the convergence threshold as a result of the damping factor tending to infinity.[7]

The Fletcher-Reeves (FR) algorithm is a conjugate gradient method which uses the previous search directions to refine the current search direction.[8] Specifically the search directions are chosen to be conjugate with respect to the Hessian matrix. For a quadratic function of N variables the algorithm converges in N steps.[8] For non-quadratic functions the FR technique is comparable to second order techniques in terms of convergence, but does not suffer from their sensitivity.

The Leapfrog Algorithm[9] is the most recently developed of the algorithms considered in this paper. It uses only gradient information to find the minimum and is thus a first order optimization technique. It simulates a particle which appears at rest at the provided starting point and is subjected to accelerations induced by the gradient of the function being minimized. It is worth noting that the acceleration induces a velocity (and therefore momentum) which allows the particle to go over 'humps' and thus finds not the nearest local minimum, but the nearest 'low local' minimum.

It is also important to scale the variables so that the gradient is equally sensitive to a step in any direction. For instance the K_3 coefficient of Brown's model[1] has to have a lower scale than the K_2 coefficient, since it works with the 6th power of the distance from the distortion centre compared to K_2's 4th power. This allows the true minimum to be found more easily.

In this paper a numerical method was deemed to have converged if the norm of the gradient was less than $\sqrt{n}10^{-6}$ (where n is the number of parameters in the function being minimized) or the step size was less than 10^{-8}. The former criterion is a function of n so that as the number of parameters increases, the required average magnitude of each element in the gradient vector remains constant.

4. HIGH ORDER BROWN MODELS

In order to show that high order Brown models are suitable for inverse distortion characterization, it is first shown that they infact provide better distortion characterization too. This was done using each of the four numerical optimization techniques discussed in Section 3 to determine the coefficients for different versions of Brown's model. This was done for each camera using 5 combinations of the 4 captured views of the optical jig portrayed in figure 1. The chosen combinations are those that have an equal amount of data in both halves of the FOV so as not to skew the results. Table 1 lists the orders of Brown's model that were evaluated. They were chosen both to represent common examples found in literature, and to investigate the different effects of determining more tangential terms, more radial terms and finding the optimal distortion centre.[10]

Table 1. Summary of distortion models evaluated

Model Number	Radial Terms	Tangential Terms	Optimal Center Found
1	1	0	N
2	1	0	Y
3	2	0	Y
4	3	2	N
5	3	2	Y
6	3	3	Y
7	5	0	N

From the above 1120 optimizations, the average level of distortion after optimization over all 32 cameras (for the combination where all 4 data sets per camera were used) is used as a measure of the effectiveness of the model/technique combination in characterizing lens distortion. This is provided in Table 2.

In order to determine if high order Brown models are stable, the standard deviations of the resultant coefficients (as scaled for optimization) for the 5 input data combinations for a camera were calculated. These were then averaged to yield a camera/technique/model robustness metric, and then averaged over the 32 cameras to create a model/technique robustness metric. This is provided in Table 3.

From Table 2 and 3 the following remarks are appropriate:[10] Levenberg-Marquardt performed poorly and consistently yielded both the poorest characterization and the largest sensitivity to input variations. This is due to the high correlations causing premature convergance due to the step-size decreasing as described in Marquardt's original derivation of the algorithm.[7]

Steepest Descent and Fletcher Reeves yielded similar results, although SD took longer to converge (as expected) and occasionally failed to converge within the maximum number of iterations, thus resulting in a higher average coefficient variance than Fletcher-Reeves. In general Fletcher-Reeves has the lowest sensitivity to input

Table 2. Mean RMS residual distortion in pixels over all 32 cameras

Model Number	Steepest Descent	Levenberg Marquardt	Fletcher Reeves	Leap frog
1	0.2810	0.3717	0.2810	0.2810
2	0.2783	0.3496	0.2865	0.2659
3	0.0765	0.4909	0.0802	0.0719
4	0.0770	0.4159	0.0758	0.0742
5	0.0754	0.3874	0.0789	0.0710
6	0.0737	0.5306	0.0749	0.0703
7	0.0771	0.4296	0.0918	0.0742

Table 3. Average coefficient variance due to input pertubations

Model Number	Steepest Descent	Levenberg Marquardt	Fletcher Reeves	Leap frog
1	0.010	0.79	0.010	0.010
2	0.800	1.40	0.490	1.40
3	0.100	1.20	0.084	0.270
4	0.041	0.75	0.066	0.097
5	0.065	0.71	0.052	0.210
6	0.069	0.63	0.056	0.260
7	0.042	0.91	0.097	0.120

variances, converged the quickest and provided a characterization second only to Leapfrog. It is thus recommended for most lens distortion applications.

Leapfrog, consistently yielded the best characterizations, which is expected as it finds the nearest "low local minimum" and not the nearest local minimum. This meant that it fitted the model better to the data provided, resulting in a higher sensitivity to input variations. Leapfrog is recommended for applications where one is certain that the captured data covers the entire FOV of the camera and is sufficiently representative of the intended operational usage of the camera.

It is also evident that adding more tangential terms, or more radial terms or determining the optimal distortion centre all improve the distortion characterization. The lenses used in this study were dominated by a second order radial copmponent but still benefitted from higher order models.

5. MODELLING INVERSE DISTORTION

During the process of lens distortion characterization, one of the end results is a set of points that it is deemed sufficiently undistorted for the application at hand. This is true whether the points are from images of known optical reference jigs or normal images where points along a straight line have been determined manually or automatically. This is because distorted points are identified and the distortion model manipulated until the the distortion metric, calculated on the candidate undistorted points, is sufficiently low.

So, since the Brown lens model is capable of modelling both barrel and pin-cushion distortion; and barrel and pin-cushion distortion can conceptually be considered inverse transformations, it is a reasonable expectation to be able to model undistortion using Brown's model.

To verify this, data was saved from the previous distortion characterization evaluation. The distorted reference points were saved together with their corresponding, supposedly undistorted, points. Only the points generated from using all four input data sets were used in this evaluation. This gave 224 data sets of varying cameras and distortion models. Two evaluations were performed: firstly, each of the distortion models in Table 1 was used to model inverse distortion for the points they generated. Secondly, the best performing algorithm was used to model the inverse distortion for the undistorted points generated by all the other models.

As the leapfrog algorithm provided the best distortion characterization results, it was used to minimize the undistortion metric (eq. (3)) by determining the optimal coefficients for Brown's model (eq. (1)). Similarly, Leapfrog's undistorted positions from the previous evaluation are the ones used for this undistortion evaluation.

For each model/technique combination the average of the undistortion metric over all 32 cameras is used as the final measure for how effective that combination is. Table 4 provides the results of the undistortion evaluations.

Table 4. Mean RMS undistortion error in pixels

Distortion	Undistortion Model	
Model	Same as distortion	Model 4
1	2.6999	0.0866
2	2.5164	0.0827
3	0.0791	0.0229
4	0.0137	0.0137
5	0.0159	0.0159
6	0.0967	0.3885
7	0.0131	0.0139

It is evident that when using the same model for both distortion and undistortion characterization, that the first order radial models perform quite poorly. There is a significant improvement when using a second order radial model, and another improvement for more complicated models. It is surprising that Model 6 performed so poorly when it yielded the best distortion characterization. This is due to it exhibiting very slow convergence and often not converging to a solution within the maximum number of iterations specified.

As Model 4 provided the best characterization of its own undistortion, it was used to model the inverse distortion for the undistorted points generated by the other models. It drastically improved upon the undistortion for the first and second order radial models, and unsurprisingly was unable to model the inverse distortion as well for the more complicated distortion models.

It has been shown that by applying sound optimization techniques, and using higher order distortion models that inverse distortion can be effectively modelled. In particular the error involved in undistortion has been reduced by 96% from the 0.42 and 0.32 pixels reported in literature[4] to 1.3 hundredths of a pixel for Model 4. This is despite the considerable distortion of up to 73 pixels inherent in the cameras used for this study.

With regards to the processing time required, only a single model call is necessary, this is equivalent to one third of an iteration in the guess-and-refine scenario. Compared to analytical undistortion models based on simplifications, it has a deterministic processing time, one fewer parameter than Mallon and Whelan's model[4] (for Model 4) and is more suitable to hardware implementation as it does not required a floating point division. The pure analytical solution, even for a single radial parameter distortion model, requires two cube roots, two square roots and 6 divisions.[11]

6. CONCLUSION

This paper addressed the issue of poor characterization of inverse distortion. It was shown that using modern numerical techniques allows higher order versions of Brown's model to be used, giving vastly improved results. Compared to current published results, the accuracy has increased 25 fold, while memory useage and processing time have been reduced. As a side effect, it was shown that the same Brown models can be used effectively to model distortion characterization too. Compared to common practice of using Levenberg-Marquardt to fit second order radial models an improvement of over 85% was obtained.

Inverse distortion modelling has been advanced significantly. Using the undistorted positions created as a side effect in the distortion characterization, it was shown that the Leapfrog algorithm can fit a high order model such that with a single model call the distorted point corresponding to an undistorted point can be found to within a few hundredths of a pixel. Low cost, low power, real-time distortion correction is now feasible.

REFERENCES

[1] Brown, D., "Decentering distortion of lenses," *Photogrammetric Engineering* **7**, 444–462 (1966).

[2] Brown, D., "Close range camera calibration," *Photogrammetric Engineering* **8**, 855–855 (1971).

[3] Conrady, A., "Decentered lens systems," *Monthly Notices of the Royal Astronomical Society* **79**, 384–390 (1919).

[4] Mallon, J. and Whelan, P., "Precise radial un-distortion of images," in [*Proceedings of the 17th International Conference on Pattern Recognition*], *ICPR 2004* **1**, 18–21 (2004).

[5] Candocia, F., "A scale-preserving lens distortion model and its application to image registration," in [*Proceedings of the 2006 Florida Conference in Recent Advances in Robotics*], *FCRAR 2006* **1**, 1–6 (2006).

[6] Levenberg, K., "A method for the solution of certain non-linear problems in least squares," *Quarterly Applied Mathematics* **2**, 164–168 (1944).

[7] Marquardt, D., "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Indust. Appl. Math.* **2**, 431–441 (1963).

[8] Fletcher, R. and Reeves, C., "Function minimization by conjugate gradients," *Computer Journal* **7**, 140–054 (1964).

[9] Snyman, J., "An improved version of the original leap-frog dynamic method for unconstrained minimization: Lfop1(b)," *Applied Mathematics and Modelling* **7**, 216–218 (1983).

[10] de Villiers, J., [*Correction of radially asymmetric lens distortion with a closed form solution and inverse function, Masters thesis*], University of Pretoria, Pretoria, RSA (2008).

[11] Cucchiara, R., Grana, C., Pratzi, A., and Vezzani, R., "A hough transform-based method for radial lens distortion correction," in [*Proceedings of the 12th International Conference on Image Analysis and Processing*], *ICIAP 2004* **1**, 182–187 (2003).