Automatic calibration and removal of distortion from scenes of structured environments

Frédéric Devernay Olivier Faugeras

INRIA Sophia Antipolis, BP 93
06902 Sophia Antipolis Cedex, France

ABSTRACT

Most algorithms in 3-D Computer Vision rely on the pinhole camera model because of its simplicity, whereas video optics, especially low-cost wide-angle lens, generate a lot of non-linear distortion which can be critical.

To find the distortion parameters of a camera, we use the following fundamental property: a camera follows the pinhole model if and only if the projection of every line in space onto the camera is a line. Consequently, if we find the transformation on the video image so that every line in space is viewed in the transformed image as a line, then we know how to remove the distortion from the image.

The algorithm consists of first doing edge extraction on a possibly distorted video sequence, then doing polygonal approximation with a large tolerance on these edges to extract possible lines from the sequence, and then finding the parameters of our distortion model that best transform these edges to segments.

Results are presented on real video images, compared with distortion calibration obtained by a full camera calibration method which uses a calibration grid.

1 INTRODUCTION

1.1 EXTERNAL, INTERNAL, AND DISTORTION CALIBRATION

In the context of 3-D computer vision, camera calibration consists of finding the mapping between the 3-D space and the camera plane. This mapping can be separated in two different transformation: first, the displacement between the origin of 3-D space and the camera coordinate system, which forms the external calibration parameters (3-D rotation and translation), and second the mapping between 3-D points in space and 2-D points on the camera plane in the camera coordinate system, which forms the internal camera calibration parameters.

The internal camera calibration parameters depend of the camera. In the case of an orthographic or affine camera model, optic rays are orthogonal to the camera plane and there are only 3 parameters corresponding to the spatial sampling of the image plane. The perspective (or projective) camera model involves two more camera parameters corresponding to the position of the principal point in the image (which is the intersection of the optical axis with the image plane). For many application which require high accuracy, or in cases where low-cost or wide-angle lens are used, the perspective model is not sufficient and more internal calibration parameters must be added to take into account camera lens distortion.

The distortion parameters are most often coupled with internal camera parameters, but we can also use a camera model in which they are decoupled. Decoupling the distortion parameters from others can be equivalent to adding more degrees of freedom to the camera model.

1.2 BRIEF SUMMARY OF EXISTING RELATED WORK

Here is an overview of the different kinds of calibration methods available. The goal of this section is not to do an extensive review, and the reader can find more information in.[13,15]

The first kind of calibration method is the one that uses a calibration grid with feature points whose world 3-D coordinates are known. These feature points, often called control points, can be corners, dots, or any features that can be easily extracted for computer images. Once the control points are identified in the image, the calibration method finds the best camera external (rotation and translation) and internal (image aspect ratio, focal length, and possibly others) parameters that correspond to the position of these points in the image. The simplest form of camera internal parameters is the standard pinhole camera,[11] but in many cases the distorsion due to wide-angle or low-quality lens has to be taken into account.[18,2] Using a calibration method with a pinhole camera model on lens with non-negligible distortion may result in high calibration errors.

The problem with these methods that compute the external and internal parameters at the same time arises from the fact that there is some kind of coupling between internal and external parameters that result in high errors on the camera internal parameters.[19]

Another family of methods is those that use geometric invariants of the image features rather than their world coordinates, like parallel lines[5,1] or the image of a sphere.[14]

The last kind of calibration techniques is those that do not need any kind of known calibration points. These are also called auto-calibration methods, and the problem with these methods is that if all the parameters of the camera are unknown, they are still very unstable.[10] Known camera motion helps in getting more stable and accurate results[17,12] but it's not always that easy to get "pure camera rotation".

A few other calibration methods are only interested in distortion calibration, like the plumb line method.[4] Another method presented in[3] uses a calibration grid to find a generic distortion function.

1.3 OVERVIEW OF OUR METHOD

Since many auto-calibration[10] or weak calibration[20] techniques rely on a pinhole (i.e. perspective) camera model, our main idea was to calibrate only the image distortion, so that any camera could be considered as a pinhole camera after the aplication of the inverse of the distortion function to image features. We also don't want to rely on a particular camera motion[17] in order to be able to work on any kind of video recordings or snapshots (e.g. surveillance video recordings) for which there can be only little knowledge on self-motion, or some observed objects may be moving.

The only constraint is that the world seen though the camera must contain 3-D lines and segments. It can be city scenes, interior scenes, or aerial views containing buildings and human-made structures. Edge extraction and polygonal approximation is performed on these images in order to detect possible 3-D edges present in the images, and after this we just look for the distortion parameters that minimize the distortion of the 3-D segments projected to the image.

After we found a first estimate of the distortion parameters, we perform another polygonal approximation on the undistorted edges, this way 3-D lines that were broken into several segments because of distortion will become one segment, and outliers (3-D curves that were detected as a 3-D segment because of their small curvature) are implicitly eliminated. We continue this iterative process until we fall into a stable minimum of the distortion error after the polygonal approximation step.

2 DESCRIPTION OF THE METHOD

2.1 THE DISTORTION MODEL

The mapping between 3-D points and 2-D image points can be decomposed into a perspective projection and a function that models the deviations from the ideal pinhole camera. A perspective projection associated with the focal length f maps a 3-D point M whose coordinates in the camera-centered coordinate system are (X, Y, Z) to an "undistorted" image point $m_u = (x_u, y_u)$ on the image plane:

$$\begin{aligned} x_u &= f\frac{X}{Z} \\ y_u &= f\frac{Y}{Z} \end{aligned} \tag{1}$$

Then, the image distortion transforms m_u to a distorted image point m_d. The image distortion model[15] is usually given as a mapping from the distorted image coordinates, which are observable in the acquired images, to the undistorted image coordinates, which are needed for further calculations.

Finally, image plane coordinates are converted to frame buffer coordinates, which can be expressed either in pixels or in normalized coordinates (i.e. pixels divided by image dimensions), depending on the unit of f:

$$\begin{aligned} x_i &= s_x x_d + c_x \\ y_i &= y_d + c_y \end{aligned} \tag{2}$$

The image distortion function can be decomposed in two terms: radial and tangential distortion. Radial distortion is a deformation of the image along the direction from a point called the center of distortion to the considered image point, and tangential distortion is a deformation perpendicular to this direction. The center of distortion is invariant under both transformations.

It was found that for many machine vision applications, tangential distortion need not to be considered.[18] The lens distortion can then be written as an infinite series:

$$x_u = x_d(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \cdots) \tag{3}$$

where $r_d = \sqrt{x_d^2 + y_d^2}$. Several tests[2,18] showed that using only the first order radial symmetric distortion parameter κ_1, one could achieve an accuracy of about 0.1 pixels in image space, using lenses exhibiting large distortion together with the other parameters of the perspective camera.[11]

In our case we want to decouple the effect of distortion from the projection on the image plane, because all we want to calibrate is the distortion. Consequently, in our model, the center of distortion (c_x, c_y) will be different from the principal point. It was shown[16] that this is mainly equivalent to adding decentering distortion terms to the distortion model of equation 3. A higher order effect of this is to apply an (very small) affine transformation to the image, but the affine transform of a pinhole camera is also a pinhole camera.

Moreover, the image aspect ratio that we use in the distortion model may not be the same as the real aspect ratio. The difference between these two aspect ratios will result in another term of tangential distortion. To summarize, κ_1 is the first order distortion, the coordinates of the center of distortion (c_x, c_y) correspond to decentering distortion because the center of distortion may be different from principal point, and the difference between the distorsion aspect ratio s_x and the real aspect ratio correspond to a term of tangential distortion.

In the following, all coordinates are frame buffer coordinates, either expressed in pixels or normalized (by dividing x by the image width and y by the image height) to be unit-less.

The undistorted coordinates are given by the formula:

$$\begin{aligned} x_u &= x_d + (x_d - c_x)\kappa_1 r_d^2 \\ y_u &= y_d + (y_d - c_y)\kappa_1 r_d^2 \end{aligned} \tag{4}$$

where $r_d = \sqrt{\left(\frac{x_d - c_x}{s_x}\right)^2 + (y_d - c_y)^2}$ is the distorted radius.

The distorted coordinates in function of the undistorted coordinated are given by the solution of the equation:

$$r_u = r_d\left(1 + \kappa_1 r_d^2\right) \tag{5}$$

where $r_u = \sqrt{\left(\frac{x_u - c_x}{s_x}\right)^2 + (y_u - c_y)^2}$ is the undistorted radius and r_d the distorted radius.

This is a polynomial of degree three of the form $r_d^3 + c r_d + d = 0$, with $c = \frac{1}{\kappa_1}$ and $d = -c r_u$, which can be solved using the Cardan method which is a direct method for solving polynomials of degree three. It has either one or three real solutions, depending on the sign of the discriminant:

$$\Delta = Q^3 + R^2$$

where $Q = \frac{c}{3}$ and $R = -\frac{d}{2}$.

If $\Delta > 0$ there is only one real solution:

$$r_d = \sqrt[3]{R + \sqrt{\Delta}} + \frac{Q}{\sqrt[3]{R + \sqrt{\Delta}}} \tag{6}$$

and if $\Delta < 0$ there are three real solutions but only one is valid because when r_u is fixed, r_d must be continuous in function of κ_1. The continuity at $\kappa 1 = 0$ gives the solution:

$$r_d = -S\cos T + S\sqrt{3}\sin T \tag{7}$$

where $S = \sqrt[3]{\sqrt{R^2 - \Delta}}$ and $T = \frac{1}{3}\arctan\frac{\sqrt{-\Delta}}{R}$

The distorted coordinates are then given by:

$$\begin{aligned} x_d &= c_x + (x_u - c_x)\frac{r_d}{r_u} \\ y_d &= c_y + (y_u - c_y)\frac{r_d}{r_u} \end{aligned} \tag{8}$$

It is cheaper in terms of calculations to detect features in the distorted image and to undistort them than to undistort the whole image (which requires solving a third degree polynomial equation at each point and bilinear interpolation to compute the gray level) and to extract the feature from the undistorted image. For some kinds of features which depend on the perspective projection and that must be detected directly in the intensity image, one must nevertheless undistort the whole image. It that case, if calibration time is not crucial but images need to be undistorted quickly, i.e. only the transform function from undistorted to distorted coordinated is to be used more often than its inverse in a program's main loop, then a good solution is to switch the distortion function and its inverse. Equation 4 would become the distortion function and equation 8 its inverse. That way the automatic distortion calibration step would be costly because it requires undistorting edge features, but one the camera is calibrated, the un-distortion of the whole intensity images would be faster.

2.2 PRINCIPLE

The goal of the distortion calibration is to find the transformation (or un-distortion) that maps the actual camera image plane onto an image following the perspective camera model. To find the distortion parameters described in section 2.1, we use the following fundamental property: 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 we need is a way to find projections of 3-D lines in the image (they are not lines anymore in the images, since they are distorted, but curves), and a way to measure how much each 3-D line is distorted in the image. Then we will just have to let vary the distortion parameters and try to minimize the distortion of edges transformed using these parameters.

2.3 EDGE DETECTION WITH SUB-PIXEL ACCURACY

The first step of the calibration consists of extracting edges from the images. Since image distortion is sometimes less than a pixel at image boundaries, there was definitely a need for an edge detection method with a sub-pixel accuracy. We developed an edge detection method[9] based on the classical Non-Maxima Suppression (NMS) of the gradient norm in the direction of the gradient which gives edge position with a precision varying from 0.05 pixels for a noise-free synthetic image to 0.3 pixels for an image Signal to Noise Ratio (SNR) of 18dB (which is actually a lot of noise, the VHS videotapes SNR is about 50dB).

2.4 FINDING 3-D SEGMENTS IN A DISTORTED IMAGE

In order to calibrate distortion, we must find edges in the image which are most probably images of 3-D segments. The goal is not to get all segments, but to find the most probable ones. For this reason, we do not care if a long segment, because of its distortion, is broken into smaller segments.

Therefore, and because we are using a subpixel edge detection method, we use a very small tolerance for polygonal approximation: the maximum distance between edge points and the segment joining both ends of the edge must typically be less than 0.4 pixels. We also put a threshold on segment length of about 60 pixels for a 640×480 image, because small segments may contain more noise than useful information about distortion.

Moreover, because of the corner rounding effect[6,7] due to edge detection, we throw out a few edgels (between 3 and 5, depending of the amount of smoothing performed on the image before edge detection) at both ends of each detected segment edge.

2.5 MEASURING DISTORTION OF A 3-D SEGMENT IN THE IMAGE

In order to find the distortion parameters we use a measure of how much each detected segment is distorted. This distortion measure will then be minimized to find the best calibration parameters. One could use for example the mean curvature of the edges, or any distance function on the edge space that would be zero if the edge is a perfect segment and the more the segment would be distorted, the bigger the distance would be.

We chose a simple measure of distortion which consists of doing a least squares approximation of each edge which should be a projection of a 3-D segment by a line,[8] and to take for the distortion error the sum of squares of the distances from the point to the line (i.e. the χ^2 of the least square approximation). That way, the error is zero if the edge lies exactly on a line, and the bigger the curvature of the edge, the bigger the distortion error. For each edge, the distortion error is defined to be:

$$\chi^2 = a\sin^2\omega - 2\,|b|\,|\sin\omega|\cos\omega + c\cos^2\omega \tag{9}$$

where:

$$a = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \tag{10}$$

$$b = \sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i \tag{11}$$

$$c = \sum_{i=1}^{n} y_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2 \tag{12}$$

$$\alpha = a - c \tag{13}$$

$$\beta = \frac{\alpha}{2\sqrt{\alpha^2 + 4b^2}} \tag{14}$$

$$|\sin\omega| = \sqrt{\frac{1}{2} - b} \tag{15}$$

$$\cos\omega = \sqrt{\frac{1}{2} + b} \tag{16}$$

ω is the angle of the line in the image, and $\sin\omega$ should have the same sign as b.

2.6 THE WHOLE CALIBRATION PROCESS

The whole distortion calibration process is not done in only one iteration (edge detection, polygonal approximation, and optimization), because there may be outliers in the segments detected by the polygonal approximation, i.e. edges which aren't really 3-D segments. Moreover, some 3-D segments may be broken into smaller edges because the first polygonal approximation is done on distorted edges. By doing another polygonal approximation after the optimization, on undistorted edges, we can eliminate many outliers easily and get longer segments which contain more information about distortion. This way we get even more accurate calibration parameters.

A first version of the calibration process is:

1. Load images.
2. Do subpixel edge detection and linking on the images.
3. Do polygonal approximation on distorted edges to extract segment candidates.
4. Compute the distortion error $E_0 = \sum \chi^2$ (sum is done over all the detected segments).
5. Optimize the distortion parameters κ_1, c_x, c_y, s_x to minimize distortion error.

6. Compute the distortion error E_1 for the optimized parameters.
7. If the relative change of error $\frac{E_0 - E_1}{E_1}$ is less than a threshold, stop here.
8. Do polygonal approximation on undistorted edges.
9. Go to step 4.

By minimizing on the four parameters when the data still contains many outliers, there is a risk of getting farther from the optimal parameters. For this reason, steps 3 to 9 are first done with optimization only on κ_1 until the termination condition of step 7 is verified, then c_x and c_y are added, and finally full optimization on the four distortion parameters is performed.

3 EXPERIMENTAL SETUP

3.1 HARDWARE

We used various hardware setups to test the accuracy of the distortion calibration, from low-cost video-conference video hardware to high-quality cameras and frame-grabber.

The lowest quality hardware is a very simple video acquisition system included with every Silicon Graphics Indy workstation. This system is not designed for accuracy nor quality and consists of an IndyCam camera coupled with the standard Vino frame grabber. The acquired image is 640×480 pixels interlaced, and contains a lot of distortion and blur caused by the cheap wide-angle lens. The use of an on-line camera allows very fast image transfer between the frame grabber and the program memory using Direct Memory Access (DMA), so that we are able to do fast distortion calibration. The quality of the whole system seems comparable to this of a VHS videotape.

Other images were acquired using an Imaging Technologies acquisition board together with several different camera setups: a Sony XC75CE camera with 8mm, 12.5mm, and 16mm lens (the smaller the focal length, the more important the distortion), and an old Pulnix TM-46 camera with 8mm lens.

3.2 SOFTWARE

The distortion calibration program is a stand-alone program that can either work on images acquired on-line using a camera and a frame grabber or acquired off-line and saved to disk. Computation of the image gradient and edge detection were done using the Robotvis libraries available at `ftp://krakatoa.inria.fr/pub/robotvis`.

The optimization step was performed using the subroutine `lmdif` from MINPACK or the subroutine `dnls1` from SLATEC, both packages are available from Netlib, at `http://www.netlib.org/`.

4 RESULTS AND COMPARISON WITH A FULL CALIBRATION METHOD

4.1 THE FULL CALIBRATION METHOD

In order to evaluate the validity of the distortion parameters obtained by our method, we compared them to those obtained by a method for full calibration (both external and internal) that incorporates comparable distortion parameters. The software we used to do full calibration implements the Tsai calibration method[18] and is freely available (the source code can be found at `ftp://ftp.teleos.com/VISION-LIST-ARCHIVE/`). This software implements calibration of external (rotation and translation) and internal camera parameters at the same time. The internal parameter set is composed of the pinhole camera parameters except for the shear parameter (which is very close to zero on CCD cameras anyway[2]), and of the first radial distorsion parameter. From the result of this calibration mechanism, we can extract the position of the principal point, the image aspect ratio, and the first radial distortion parameter.

As seen in section 2.1, though, these are not exactly the same parameters as those that we can compute using our method, since we allow more degrees of freedom for the distortion function: two more parameters of decentering distortion and one parameter of tangential distortion. Having different coordinates for the principal point and the center of distortion, and for the image aspect ratio and distortion aspect ratio.

4.2 RESULTS

The distortion calibration method was applied to sets of about 30 images (see Figure 1) for each camera/lens combination, and the results for the four parameters of distortion are shown in Table 1. The initial values for the distortion parameters before the optimization were set to "reasonable" values, i.e. the center of distortion was set to the center of the image, κ_1 was set to zero, and s_x to the image aspect ratio, computed for the camera specifications. For the IndyCam, this gave $c_x = c_y = \frac{1}{2}$, $\kappa_1 = 0$ and $s_x = \frac{3}{4}$.

Figure 1: Some of the images that were used for distortion calibration.

Figure 2 shows a sample image, before and after the correction. This image was affected by pin-cushion distortion, corresponding to a positive value of κ_1. Barrel distortion corresponds to negative values of κ_1.

We also calibrated the same cameras using the Tsai method and a calibration grid (Figure 3) with 128 points, and we computed some parameters corresponding more or less to our distortion parameters from the result of this full calibration method (Table 2). As explained in section 2.1, these are not the same as our distortion parameters, because we introduced a few more degrees of freedom in the distortion function, allowing decentering and tangential distortion. This explains why the distortion center found on low-distortion cameras such as the Sony 16mm are so far away from the principal point.

camera/lens	IndyCam	Sony 8mm	Sony 12.5mm	Sony 16mm	Pulnix 8mm
κ_1	0.154	0.0412	0.0164	0.0117	-0.0415
c_x	0.493	0.635	0.518	0.408	0.496
c_y	0.503	0.405	0.122	0.205	0.490
s_x	0.738	0.619	0.689	0.663	0.590

Table 1: The distortion parameters obtained on various camera/lens setups using our method, in normalized image coordinates: First radial distortion parameter, position of the center of distortion, and distortion aspect ratio.

Figure 2: A distorted image with the detected segments (left) and the same image at the end of the distortion calibration with segments extracted from undistorted edges (right): some outliers were removed and longer segments are detected.

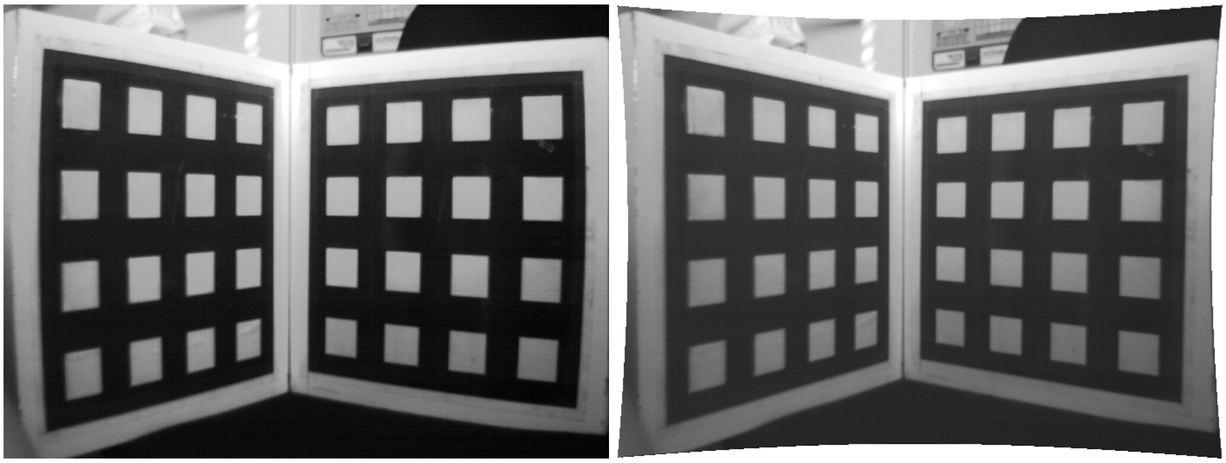

Figure 3: The calibration grid used for Tsai calibration: original distorted image (left) and image undistorted using the parameters computed by our method (right).

camera/lens	IndyCam	Sony 8mm	Sony 12.5mm	Sony 16mm
κ_1	0.135	0.0358	0.00772	0.00375
c_x	0.475	0.514	0.498	0.484
c_y	0.503	0.476	0.501	0.487
s_x	0.732	0.678	0.679	0.678

Table 2: The distortion parameters obtained using the Tsai calibration method, in normalized image coordinates: First radial distortion parameter, position of the principal point, and image aspect ratio. They do not have the same meaning as in Table 1, as explained in section 2.1.

For cameras with high distortion, like the IndyCam and the cameras with 8mm lens, The center of distortion and the distortion aspect ratio are close to the principal point and the image aspect ratio. For better comparison with a grid-based calibration, we should use a camera model that includes the same distortion parameters as ours, i.e. introduce the center of distortion and distortion aspect ratio.

5 DISCUSSION

With computer vision applications demanding more and more accuracy in the camera model and the calibration of its parameters, there is definitely a need for calibration methods that don't rely on the simple and linear pinhole camera model. Camera optics still have lots of distortion, and zero-distortion wide-angle lens exist but remain very expensive.

The automatic distortion calibration method presented here has many advantages over other existing calibration methods that use a camera model with distortion.[2,3,17,18] First, it makes very few assumptions on the observed world: there is no need for a calibration grid.[2,3,18] All it needs is images of scenes containing 3-D segments, like interior scenes or city scenes. Second, it is completely automatic, and camera motion need not to be known.[16,17] It can be applied to images acquired off-line, as an example they could come from a surveillance videotape or a portative camcorder. The comparison with a full grid-based calibration method presented in this paper is still not complete, since the grid-based calibration method we used didn't have the same distortion parameters as our method. The next step in the validation of this method is to do better comparisons.

Once the distortion is calibrated, any computer vision algorithm that relies on the pinhole camera model can be used, simply by applying the inverse of the distortion either to image features (edges, corners, etc.) or to the whole image. This method could also be used together with auto-calibration or weak calibration methods that would take into account the distortion parameters. The distortion calibration could be done before auto-calibration, so that the latter would use un-distorted features and images, or during auto-calibration, the distortion error being taken into account during the auto-calibration process.

6 REFERENCES

[1] P. Beardsley, D. Murray, and A. Zissermann. Camera calibration using multiple images. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision*, pages 312–320, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[2] Horst A. Beyer. Accurate calibration of CCD-cameras. In *Proceedings of the International Conference on*

Computer Vision and Pattern Recognition, Urbana Champaign, IL, June 1992. IEEE.

[3] P. Brand, R. Mohr, and P. Bobet. Distorsions optiques : correction dans un modèle projectif. Technical Report 1933, LIFIA–INRIA Rhône-Alpes, 1993.

[4] Duane C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[5] Bruno Caprile and Vincent Torre. Using Vanishing Points for Camera Calibration. *The International Journal of Computer Vision*, 4(2):127–140, March 1990.

[6] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *Proceedings of the 3rd Proc. International Conference on Computer Vision*, pages 66–70, Osaka, Japan, December 1990. IEEE Computer Society Press.

[7] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *The International Journal of Computer Vision*, 10(2):101–124, 1993.

[8] Rachid Deriche, Régis Vaillant, and Olivier Faugeras. From noisy edge points to 3D reconstruction of a scene: A robust approach and its uncertainty analysis. In *Proceedings of the 7th Scandinavian Conference on Image Analysis*, pages 225–232, Alborg, Denmark, August 1991.

[9] Frédéric Devernay. A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical report, INRIA, 1995.

[10] Olivier Faugeras, Tuan Luong, and Steven Maybank. Camera self-calibration: theory and experiments. In Giulio Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision*, number 588 in Lecture Notes in Computer Science, pages 321–334, Santa-Margherita, Italy, May 1992. Springer-Verlag.

[11] Olivier Faugeras and Giorgio Toscani. Structure from Motion using the Reconstruction and Reprojection Technique. In *IEEE Workshop on Computer Vision*, pages 345–348, Miami Beach, November-December 1987. IEEE Computer Society.

[12] Richard Hartley. Self-calibration from multiple views with a rotating camera. In J-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, pages 471–478, Stockholm, Sweden, May 1994. Springer-Verlag.

[13] R. Lenz and R. Tsai. Techniques for calibrating of the scale factor and image center for high accuracy 3D machine vision metrology. In *International Conference on Robotics and Automation*, pages 68–75, Raleigh, NC, 1987.

[14] M. A. Penna. Camera calibration: A quick and easy way to determine the scale factor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1240–1245, 1991.

[15] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.

[16] Gideon P. Stein. Internal camera calibration using rotation and geometric shapes. Master's thesis, Massachusetts Institute of Technology, June 1993. AITR-1426.

[17] Gideon P. Stein. Accurate internal camera calibration using rotation with analysis of sources of error. In *Proceedings of the 5th Proc. International Conference on Computer Vision*, Boston, MA, June 1995. IEEE Computer Society Press.

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

[19] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.

[20] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Research Report 2273, INRIA Sophia-Antipolis, France, May 1994. submitted to *Artificial Intelligence Journal*.