Sub-pixel Accuracy Corner Detection

E. Hammer Johnson
Saint Olaf College
johnsoeh@stolaf.edu

I. Motivation

The parameters of most camera calibration models are determined by ideal–distorted pairs of image coordinates. Whereas the ideal coordinates of these reference points are assumed *a priori*, the corresponding points in the distorted image—often situated as corners on a checkerboard—must be located. In [1], Zhang simulates camera calibration and applies both random noise and systematic nonplanarity to points on the calibration target: systematic bias of reference points caused absolute error of more than 800% that caused by Gaussian noise. Thus in reference point location, accuracy is paramount and precision of secondary concern.

This paper reports on calibration work done in collaboration with Tim Yates (yates@stolaf.edu) and Olaf Hall-Holt (olaf@stolaf.edu) at Saint Olaf College. In particular, we hope to calibrate with sub-pixel accuracy a stereo pair of cameras and thereby increase the quality of three-dimensional models produced from their images. Calibration work is restricted to image sets collected in previous work; each such image contains two LCD screens, each displaying a checkerboard and positioned in fixed but unknown relation to each other, and a dark background that will minimally interfere with feature detection. We divide the task of sub-pixel corner detection into *corner approximation* (Section II) and *corner refinement* (Section III) and provide implementations of each. Section IV presents results of calibration with our methods, which are then compared to other work in Section V. In all cases, “corner” abbreviates “interior corner.”

II. Corner Approximation

Given an image containing two calibration boards, we seek to (i) identify the pose and dimensions of each board and (ii) obtain for each an ordered set of image coordinates which approximate the individual locations of each corner.

A. ChESS Indicator

We implement the ChESS feature detector as described in [2]. The ChESS response at each point in an image gives a positive or even negative number, the likelihood of that point being a corner on a checkerboard. To that end, the ChESS detector considers the intensities of pixels in a circular sample window around the point in question at a radius of approximately 5 pixels (Fig. 1).

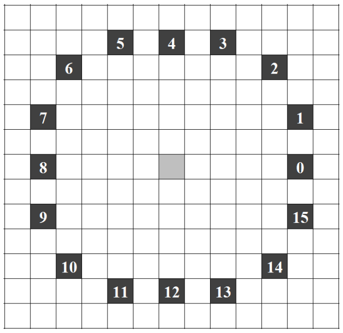

Figure 1. ChESS sample window for light grey pixel in center.

In practice, we find that ChESS performance on our data set does not benefit from calculation of *mean response*; we therefore omit *mean response* altogether in order to reduce runtime complexity. Denoting the intensities of pixels around the circular sample window as $I_0,\ I_1,\ \ldots,\ I_{14},\ I_{15}$ (Fig. 1), we calculate

$$\textit{response} = \textit{sum response} - \textit{diff response}$$

where

$$\textit{sum response} = \sum_{n=0}^{3} |(I_n + I_{n+8}) - (I_{n+4} + I_{n+12})|$$

and

$$\textit{diff response} = \sum_{n=0}^{7} |I_n - I_{n+8}|.$$

It is worth noting that each of these responses has a maximum potential value of eight times maximum potential pixel intensity. For more information the interested reader is referred to [2].

B. Board Identification

In practice, we find the intensity of ChESS response to be effectively constant across corners on a given board but encounter significant differences in intensity between boards due to variation in board orientation and contrast in the original image. Additionally, the spatial characteristics of corner sets may of course also differ between boards. Analyzing the ChESS response of each board separately thus admits much more aggressive filtering methods.

We use a variation on 2-means clustering [3] to partition each ChESS response image into two boards. In order to satisfactorily cluster our continuous image data, clustering is weighted by intensity of ChESS response. In particular, each pixel at location (x, y) with ChESS response z has the effect of z samples at (x, y) in a generic k-means implementation. As our boards always appear in horizontal sequence, we seed 2-means with the minimum and maximum x coordinates of all pixels with significant or "non-noise" ChESS response. The image distance between our two boards allows quite liberal 2-means stopping conditions, and in practice we achieve convergence after only two clustering iterations.

C. Corner Location

Given the ChESS response of an image segment containing a single calibration board, we produce an unordered set of image coordinates which contains exactly one point per real-world board corner. We use three criteria to filter ChESS responses.

1) Intensity: Our ChESS response typically includes slight background signal due to noise in the original images. A high-pass filter by pixel intensity applied relative to maximum pixel intensity easily flattens this background. We retain all points with ChESS response at least 20% that of the maximum observed.

2) Local maximality: As shown in [2], typical ChESS response patterns for corners span many pixels and peak at the center of intersection. Modified to break ties between equal-and-neighboring maxima, non-maximum suppression in a window size comparable to that used by ChESS replaces true positives with a relatively accurate single point approximation. As a secondary benefit, this also severely limits the density of false positives.

3) Distance to nearest neighbor: False positive ChESS responses are likely to lack the regular nearby responses characteristic of true checkerboard corners. Applied after non-maximal suppression, a band-pass filter by distance to nearest neighbor removes such "geographic outliers" as long as true corners are far more common than false positives. Our data require only weak geographic filtering: we retain all points with nearest neighbor at a distance within a factor of two of the arithmetic mean of all nearest neighbor distances.

The preceding three criteria satisfactorily filter our data set. In other circumstances, the addition of more advanced filtering strategies such as those suggested in [2] could further increase resilience to noise.

D. Corner Identification

Given an unordered set of image coordinates representing corners on a calibration board, we determine the dimensions of the board and define a one-to-one correspondence between image coordinates and board coordinates. It is worth noting that the product of board dimensions is assumed to be exactly the number of image coordinates in the given set.

1) Four corners: We first identify the four corners of the board in order to determine its pose. For all 4-permutations (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) of the given image coordinates, we compute

$$\frac{1}{2}\sum_{i=0}^{3}(x_i y_{i+1} - x_{i+1} y_i)$$

where $n \in \mathbb{Z}_4$ for all (x_n, y_n). In the case that a permutation describes a simple polygon, this sum represents its signed area; if a complex polygon, the sum lies between 0 and the correct signed area. Thus a 4-permutation with so-computed maximal "area" contains the four board corners in counterclockwise order. It is worth noting that in a displayed image this order will *appear* to be clockwise.

Four equivalent so-maximal permutations always exist. For consistency, we order corners beginning with that nearest the origin according to taxicab geometry; this has the intuitive effect of selecting the "upper-left-most" corner in a displayed image.

2) Homography and dimensions: As detailed in [1], these four corners determine up to a scale factor a homography from board coordinates to image coordinates. We then speculatively vary the dimensions of the pre-image board across all co-divisors of the total number of given image coordinates. For each potential board dimension, we associate the homography image of each corner from the pre-image board with the nearest given image coordinate. We choose the dimension with the least accumulated error between homography images and given image coordinates.

The preceding approach orders corners satisfactorily as long as (i) displacement from camera distortion never exceeds half the image coordinate side length of any calibration target square, (ii) filtering as before can produce noiseless input data, and (iii) polynomial runtime (naively degree 4) with the number of corners per board is feasible.

III. Corner Refinement

Given a calibation image and an ordered set of corner coordinates for each board in it, we seek to refine the given corner locations to sub-pixel accuracy and give independent results for each image color channel. As a first step toward sub-pixel accuracy corner refinement, we implement methods from [4] with several slight modifications.

A. Sub-pixel Edge Detection

Given a calibration image and several parameters, we detect sub-pixel edge locations as in [4]. Considering change in pixel intensity along a scanline or scancolumn, an edge is identified by (i) a positive change in intensity greater than a given threshold followed by (ii) monotonic transition to (iii) a negative change in intensity less than a second threshold, or the reverse of this sequence. A line is fitted to the monotonic sequence from (ii), a "gray value" is estimated as the midpoint between the averages of pixel intensities within a short distance of either side of the sequence, and a sub-pixel edge location is detected at the intersection of the fitted line and estimated gray value.

B. Sub-pixel Corner Detection

Given a set of sub-pixel edge points detected as above in a calibration image along with information for the same image from our corner approximation methods, we seek to refine approximate corner locations to sub-pixel accuracy.

1) Edge identification: As a modification to methods in [4], we use the inverse of the homography computed during corner approximation to associate detected edge points to edges on a calibration board. In particular, we compute the three-dimensional pre-image of each two-dimensional sub-pixel edge location and associate the given edge point with the nearest edge on the pre-image board; comparison of distances in three-dimensional space avoids the distortion of two-dimensional perspective.

2) Edge reconstruction: Returning to image coordinates, we fit a line to each set of sub-pixel edge locations associated with a single board edge; compared to association with (all) “near” lines in [4], association with (only) “nearest” line increases signal-to-noise ratio in this step and allows a single rather than iterative computation. As a final deviation from [4], we fit lines using Deming regression [5] rather than simple linear regression as either coordinate in detected edge locations may err.

3) Corner reconstruction: We place sub-pixel corner locations at the intersection of the two relevant reconstructed edges as in [4].

IV. Results

We present preliminary results from our work.

A. Approximation

Our corner approximation methods are tailored to our data set; the following discussion addresses questions of wider applicability.

Accuracy and precision: The ChESS response pattern at and around a corner point varies by corner orientation and thus closely resembles the response pattern of other corners on the same board [2]. This confuses the relationship between ChESS accuracy and precision. We observe absolute error of 1–2 pixels when corners are placed at local ChESS maxima and approximately 0.5 pixels when placed at local “centers of mass” [2]. For an extensive analysis of ChESS performance, the interested reader is referred to [2].

Robustness: Our calibration images contain modest amounts of noise caused by LCD screen powerlights and screen glare. In some cases, a powerlight augmented by diagonal glare produces an effective checkerboard corner. Our corner approximation methods overcome this noise to correctly label all corners in all of our images. As previously indicated, our simple corner approximation approach could be applied to noisier data sets after the addition of more advanced filtering strategies such as those suggested in [2] or which consider more information from the original image such as color.

B. Refinement

Our corner refinement methods do not currently improve calibration results over those using approximate locations. The following discussion addresses potential performance.

Accuracy and precision: Our corner refinement methods ensure the collinearity of board corners in image coordinates; thus the accuracy of sub-pixel corner locations necessarily worsens as distortion across a given board increases. Future work could improve accuracy by fitting lines to the edges of individual squares rather than those spanning entire checkerboards [4].

A lower bound on the precision of refined corner locations could be derived from measures of error during Deming regression. Calculating the variance of distance between detected sub-pixel edge points and associated lines may be of interest when calibration results with refined corner locations surpass those using approximate locations.

Robustness: As in [4], our corner refinement methods require manual tuning of parameters and still do not satisfactorily overcome variation in brightness and contrast across a given image. Future work could implement unparameterized subpixel edge detection methods such as those in [6], [7].

V. Related Work

A. *Saint Olaf Advanced Team Project* [4]

Our work builds upon and hopes to improve methods developed in earlier coursework by Brown, et al. In particular, we automate the corner approximation process, automate detection of checkerboard dimensions, and reduce noise levels in edge data from which corner locations are refined.

B. *OpenCV* [8]

We find that OpenCV approximate corner detection is particularly robust to noise in the background of a high-quality calibration image but struggles to identify corners on a checkerboard with low contrast or extreme perspective. Our calibration images and one of the two boards in particular consistently exhibit both of these properties, so little comparison can be drawn between OpenCV and our methods in the context of this data set.

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.888718

[2] S. Bennett and J. Lasenby, "ChESS: Quick and robust detection of chess-board features," *University of Cambridge Signal Processing and Communications Laboratory*, 2011. [Online]. Available: http://www-sigproc.eng.cam.ac.uk/~sb476/publications/ChESS.pdf

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66)*. Berkeley, Calif.: Univ. California Press, 1967, pp. Vol. I: Statistics, pp. 281–297.

[4] J. Brown, R. Magee, and L. Roth, "Refinement of plane based calibration through enhanced precision of feature detection," *Saint Olaf Advanced Team Project*, 2012.

[5] W. Deming, *Statistical adjustment of data*. New York: Dover, 1943.

[6] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jun. 1986. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.1986.4767851

[7] P. Pelgrims, G. V. de Velde, and B. V. de Vondel, "Sub-pixel edge detection," *De Nayer Instituut*, 2004. [Online]. Available: emsys.denayer.wenk.be/emcam/subpix_eng.pdf

[8] [Online]. Available: http://opencv.willowgarage.com/wiki/