Corner Detection and its Application to Camera Calibration

Mark A. Livingston and H. Harlyn Baker

Hewlett-Packard Laboratories, Palo Alto, California

Abstract

We present a slightly modified scale space corner detection algorithm. We then describe refinement methods for the localized features. One refinement operates in scale space, the other along the image gradient. Finally, we summzrize the results of these algorithms towards improving camera calibration performance.

1 Introduction

We began our work with three observations.

1. Detection of features such as corners is often the first step in camera calibration methods [21, 3].
2. Camera calibration from point correspondences is quite sensitive to noise in the coordinates of the points [15, 2].
3. While much work has been to characterize and reduce the sensitivity of calibration methods [1, 13, 7], it would improve the calibration if we can improve the performance of the raw feature detectors [8].

Given these observations of our own work, we investigate corner detection and ways to improve it. We then conduct an experiment on the effect we have on the camera calibration methods.

2 Related Work

2.1 Corner Detection

While there are many methods for corner detection, we focus this brief summary on methods that use intensity gradients and only mention recent methods, such as those based on topographic analysis [12] or morphological operators [9]. Broad reviews [18, 17, 5] and more detailed analysis of image geometry techniques [22] are available.

The corner detectors with which we have been working use image intensity derivatives to determine what the "cornerness" of a point is. A corner should have high derivatives in both directions along the pixel grid. The Plessey operator [6] is one commonly-used example of such an operator, and was the starting point for our work. Similar operators have frequently been seen in the literature [22, 20, 4].

The Plessey operator works by finding the first derivative in each of the image axis directions, which we will refer to as x and y. The following matrix is then constructed.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = \begin{bmatrix} \left(\frac{\delta I}{\delta x}\right)^2 & \left(\frac{\delta I}{\delta x}\right)\left(\frac{\delta I}{\delta y}\right) \\ \left(\frac{\delta I}{\delta x}\right)\left(\frac{\delta I}{\delta y}\right) & \left(\frac{\delta I}{\delta y}\right)^2 \end{bmatrix}$$

The response function is then given as

$$R = \det M - k\text{trace}M. \tag{1}$$

Previous authors have found this operator to be good at detecting corners, but not as accurate as desired at localizing the corners [23, 22].

Still others use second derivatives [20]. Scale-space detection techniques for edges and corners also have a prominent place in the literature [10, 16, 14]. Searching in scale space implies using a varying window size over which to approximate the derivatives. Wang and Brady [20] noted that really one would like to use an infinitely small window, which can be apporximated by extrapolating from a series of smaller and smaller windows. The trade-off is that smaller windows are more susceptible to image noise.

2.2 Camera Calibration

Camera calibration is a large topic and we will again give only the brief treatment necessary to set up our results. Tsai's algorithm [19] uses image sightings of known 3D points to constrain both intrinsic and extrinsic parameters of the camera. More recent work has focused on derivatives of the Eight Point Algorithm [11], which uses only corresponding image points. The original algorithm was generally found to be quite susceptible to noise, although this problem can be reduced with normalizing transformations [7]. The original algorithm assumed the camera had known intrinsic parameters; extensions erased this assumption to form the fundamental matrix [13], which is a homography between two image planes. Much work [13, 1] has been devoted to analyzing the stability and error in the fundamental matrix. While this work is certainly important, it should be self-evident that the better the image correspondence data used by any algorithm to compute the fundamental matrix is, the better the answer the algorithm will compute.

It has been shown [8] that for circular landmarks, a perspective distortion can move the projection of the center of the landmark away from the centroid of the pixels that contain part of the landmark. It is this type of error that we hope to eliminate for systems that rely on corner detectors, and to demonstrate improvement in the camera calibration.

3 Scale-Space Feature Detection

3.1 Single-scale operator

As noted above, we began our work by using the Plessey operator to detect corners. Looking again at Equation 1, we note that the determinant of the matrix contains the negative of the product of the two image derivatives. If both these values are large, this will push the operator value towards the negative, which is not supposed to indicate an edge. Our first change, then, is to simply change the sign, so that we no longer use the determinant of the matrix.

$$R = AB + C^2 - k\text{trace}M. \tag{2}$$

where A, B, and C are defined as in Equation 1. This operator performed well on images of our calibration target

(Figure 1).

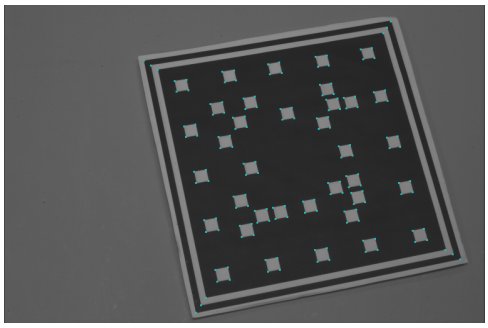

Figure 1: Our calibration target, with corners detected.

The operator is computed on the pixel grid and then interpolated to achieve sub-pixel resolution.

3.2 Localization in Scale Space

Following the lead of Wang and Brady [20], we apply this operator at scales of 3×3, 5×5, ... 11×11. We then find a least-squares quadratic curve in scale space to compute (separately) the estimated location for the x and y coordinates of the corner for a single-pixel window (SSE-1 algorithm) and for an infinitessimally small window (SSE-0 algorithm).

Working in scale-space simulation with ray traced images, we notice a correlation between the error from the detected to the ideal corners and the image gradient. Not surprisingly, the larger the window, the more strongly correlated the gradient and the error vector were. As the window reduces to 3×3, the correlation is still evident, but not overwhelming. For the 11×11 windows, all 144 test points in our simulation had a dot product of at least 0.9 between their error vector and the image gradient. For the 3×3 window, about 45% had a dot product in that range, while the remaining points were spread over the range of 0.35-0.90 (Figure 2). Keep in mind that these images are nearly noiseless, as the ray traced simulation does not include adding random noise. Whether such a correlation exists in real images, we can only guess. However, this does give rise to an efficient search algorithm to attempt correction of the feature detector.

0.35–0.40	xx
0.40–0.45	xx
0.45–0.50	xx
0.50–0.55	xx
0.55–0.60	xx
0.60–0.65	xxxxxxx
0.65–0.70	x
0.70–0.75	xxxxxxxxx
0.75–0.80	xxxxx
0.80–0.85	xxxxx
0.85–0.90	xxxxxx
0.90–0.95	xxxxx
0.95–1.00	xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Figure 2: Histogram of the dot product between the error vector from the detected to the ideal landmark and the image gradient. This shows a significant correlation between the two vectors. This histogram contains all the test points, even though it covers less than half the range of the dot product. Each `x` represents two sample points.

4 Refinement of Feature Localization

Our goal in this work was to see what effect the extrapolation in scale space would have on the camera calibration. For calibration, we manually construct a 2D model for the target, print it, and then take eight photographs. We then feed the known 2D model and the detected corners into Zhang's calibration method [21]. This method is a derivative of the fundamental matrix approach. We also search along the image gradient for a single-parameter transformation of the detected features to see if this improves our calibration.

4.1 Metric for Camera Calibration

First, we need a metric for camera calibration. In the absence of any ground truth, we use the following self-consistency metric to measure the success of our calibration. Note that the input information is only the inpyut coordinates of the detected features. While it is true that when using Zhang's calibration method, we could enforce a coordinate system on the target and derive known 3D coordinates, this is not true of all calibration methods, so we prefer to avoid this assumption.

The image coordinates and the camera calibration enable us to compute a 3D location for each calibration point in the target. We use the point that minimizes the sum of squared distances between the point estimate and the rays which are given as corresponding for each point (one ray per point from each image). A metric of camera calibration error, then, is the average or maximum distance from any computed 3D point to any of its formative rays.

We then project these 3D points back onto the images. We then have a 2D error metric: distance from the projected point to the detected feature location. We can consider the average and maximum values.

4.2 Equipment

We did this for two cameras. First was a Kodak DCS460, which has a 6 Mpix grayscale sensor behind a Bayer filter color mosaic. We demosaic the images to a grayscale image, then for reduced memory consumption in our end applications, we typically reduce the resolution by a factor of two in both dimensions, leaving us with a 768×512 grayscale image for calibration. (Our applications use color, but we find that demosaic error interfere with calibration and color adds nothing to the calibration process.) These images are relatively free of noise., as you can see in Figure 1.

The second camera was a DragonFly Digital Video Camera from Point Grey research `http://www.ptgrey.com`. We use this in VGA mode, capturing 640x480 frames. We captured at 7.5Hz in this experiment.

4.3 Calibration improvements

The results from our experiments are summarized in Table 1.

The DCS460 has very little noise, so as Wang and Brady predict, there is very little to be gained by using extrapolation in scale space. In fact, as you can see in the table, it actually increases the calibration error. This is rather surprising, but given that the 3×3 window operator yielded a calibration with under 1.0 pixels of maximum reprojection error, we can hardly expect significant improvement.

We search the gradient at a resolution of 0.05 times its real length. The gradient method yielded a slight improvement to the calibration. Upon seeing this result, we opted to try

Algorithm	DCS460	DragonFly
Raw 5x5	1.019625	1.399554
Raw 3x3	0.956231	2.279644
SSE-1	1.380844	1.706845
SSE-0	1.805181	2.496607
Grad	0.954192, $t = 1.0$	1.349952, $t = 1.9$
Conj	0.908742, $t = 1.3$	1.388987, $t = 1.9$

Table 1: Results from our experiments to improve camera calibration by improving feature localization accuracy. The error metric used here is the 2D metric described in Section 4.1, with a value in pixels. However, only the maximum error is shown. The average error shows similar results, although not always at the same location along the gradient or conjugate gradient direction. We are still investigating these methods. Note that the image plane of the Kodak camera is subsampled to 768×512 and the DragonFly camera captures VGA-resolution video frames.

the conjugate gradient, shown in the last line of the table. It is a little surprising that the conjugate gradient search found a better solution than the gradient search, since our simulations showed that the correlation was strongest between the gradient direction and the error in the feature detector. We are still investigating this result. Both improvements are modest for the DCS460. The table gives the parameter t that measures the length of the vector in the direction of the gradient or conjugate gradient that produced the best result. We show only maximum error in the table, and the average error shows similar results. The parameter t is not the same for the average, and the search space is very flat.

For the DragonFly, we see that the 3×3 window performs more poorly than the larger windows, an indicator that image noise is causing more difficulty for the calibration process. It also implies that our extrapolation process should rely less on the detected features from this small window size. We thus extrapolate from only the 5×5 and larger window sizes when running the SSE-1 version of the algorithm. The problem this creates, of course, is that we are extrapolating to well outside the range of the original data, and the approximation is less certain, as evidenced by the performance.

The gradient search and conjugate gradient searches again found a minor improvement over the best windowed search. It is interesting to note that in this noisier image, the gradient search did better than the conjugate gradient.

5 Summary

We have attempted two algorithms (plus variations) to improve camera calibration performance by attempting to refine localized feature coordinates. The scale space algorithm has yet to demonstrate improvement, perhaps needing more intelligence to handle noisy image data. The (conjugate) gradient-based search method seems to show some promise, although it is unclear why the improvements come in the conjugate gradient direction. Still, this seems a promising direction for further research.

References

[1] Csurka, G., Zeller, C., Zhang, Z., and Faugeras, O. D. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding 68*, 1 (Oct. 1997), 18–36.

[2] Faugeras, O. D., Luong, Q.-T., and Maybank, S. J. Camera self-calibration: Theory and experiments. In *Proceedings of the Second European Conference on Computer Vision (ECCV '92)* (May 1992), pp. 321–334.

[3] Fitzgibbon, A. W., and Zisserman, A. Automatic camera recovery for closed or open image sequences. In *Proceedings of European Conference on Computer Vision (ECCV 1998)* (June 1998), pp. 311–326.

[4] Fö"rstner, W. A feature based correspondences algorithm for image matching. *Intl. Arch. Photogrammetric Remote Sensing 26* (1986), 150–166.

[5] Haralick, R. M., and Shapiro, L. G. *Computer and Robot Vision*. Addison-Wesley, Reading, Mass., USA, 1992.

[6] Harris, C., and Stephens, M. A combined corner and edge detector. In *Fourth Alvey Vision Conference* (1988), pp. 147–151.

[7] Hartley, R. I. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision* (June 1995), IEEE, pp. 1064–1070.

[8] Heikkilä, J., and Silvén, O. A four-step camera calibration procedure with implicit image correction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)* (1997), pp. 1106–1112.

[9] Laganière, R. A morphological operator for corner detection. *Pattern Recognition 31*, 11 (Nov. 1998), 1643–1652.

[10] Lindeberg, T. *Principles for Automatic Scale Selection*. Academic Press, Mar. 1999, pp. 239–274.

[11] Longuet-Higgins, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature 293* (Sept. 1981), 133–135.

[12] Luo, B., Cross, A. D. J., and Hancock, E. Corner detection via topographic analysis of vector-potential. *Pattern Recognition Letters 20*, 6 (June 1999), 635–650.

[13] Luong, Q.-T., and Faugeras, O. D. Camera calibration, scene motion and structure recovery from point correspondences and fundamental matrices. *The International Journal of Computer Vision 22*, 3 (1997), 261–289.

[14] Mokhtarian, F., and Suomela, R. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 12 (Dec. 1998), 1376–1381.

[15] Oliensis, J. A critique of structure-from-motion algorithms. *Computer Vision and Image Understanding 80*, 2 (Nov. 2000), 172–214.

[16] Park, D. J., Nam, K. M., and Hong Park, R. Multiresolution edge detection techniques. *Pattern Recognition 28*, 2 (Feb. 1995), 211–229.

[17] Rohr, K. Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision 4*, 2 (1994), 139–150.

[18] Smith, S. M. Reviews of optic flow, motion segmentation, edge finding, and corner finding. Tech. Rep. TR97SMS1, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, Oxford University, 1997. `http://www.fmrib.ox.ac.uk/~steve/review/`.

[19] Tsai, R. Y. 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 RA-3*, 4 (Aug. 1987), 323–344.

[20] Wang, H., and Brady, M. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing 13*, 9 (Nov. 1995), 695–703.

[21] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 11 (Nov. 2000), 1330–1334.

[22] Zheng, Z., Wang, H., and Teoh, E. K. Analysis of gray level corner detection. *Pattern Recognition Letters 20*, 2 (Feb. 1999), 149–162.

[23] Zitová, B., Flusser, J., Kautsky, J., and Peters, G. Feature point detection in multiframe images. In *Czech Pattern Recognition Workshop 2000*.