

Griffith Research Online

<https://research-repository.griffith.edu.au>

Pattern-Based Corner Detection Algorithm

Author

Le, Xuesong, Gonzalez, Ruben

Published

2009

Conference Title

ISPA 2009: Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis ISPA 2009

DOI

<https://doi.org/10.1109/ISPA.2009.5297715>

Copyright Statement

Copyright 2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Downloaded from

<http://hdl.handle.net/10072/29961>

Pattern-Based Corner Detection Algorithm

Xuesong Le ^{#1}, Ruben Gonzalez ^{*2}

*# School of Information, communication, and Technology, Griffith University
Griffith University, Parklands Drive, Southport QLD 4222 Australia*

¹ x.le@griffith.edu.au

** NICTA, Staff House Road, St Lucia QLD and Institute for Intelligent Integrated Systems
Griffith University, PMB 50, Gold Coast Mail Centre, QLD, 4217*

² r.gonzalez@griffith.edu.au

Abstract

In this paper, the problem of detecting corners is simplified into detecting geometric patterns in local neighborhood regions. This approach outperforms Harris corner detector, SIFT, FAST corner detector in terms of localisation, rotation invariant and noise robustness, invariant to luminance changes under common conditions.

1. Introduction

Corners are special features in images. They are of great use in computing the optimal flow and structure from motion. The majority of corner detection algorithms [1] work by defining a corner response function whose local response attains a maximum so that corner can be identified. This quantitative measure of relative strength of being a corner point is called the "cornerness value" of the point.

Many corner detectors developed recently can be divided into two main categories, correlation-based and geometrical-based. Moravec [2] computes the sum-of-squared-differences (SSE) between a window around a candidate corner and windows shifted by a small amount in four directions. A corner can be detected by find the minimum SSE above a threshold value produced by any of the four shifts. Many of the weaknesses of the Moravec operator are addressed by Harris operator [3]. For example, as the variation was computed by Moravec along four directions only, this operator is sensitive to strong edges under certain directions. Harris operator is generally considered to be the best operator with respect to detect true corners, but has poor localization and is expensive to compute. Zheng and Wang [4] have proposed a computationally simplified cornerness measure by calculating two second-order gradient-multiple images instead of three with slight reduction in computation in performance. The Forstner operator [5] uses a similar measure of cornerness to the Harris operator and uses the local statistics to calculate the selection threshold. Besides being time consuming, they are prone to poorly perform in real images as they assume uniform grey level within

regions. Scale invariant feature transform was proposed by David Lowe [6]. Lowe's algorithm finds stable features over scale space by repeatedly detecting minima and maxima over difference-of-Gaussian images. At each of these minima and maxima, a detailed model is fit to determine location, scale and contrast. Once a stable feature has been detected, gradient orientation is assigned and a key point descriptor vector is formed. It has the advantage of robustness to changes in illumination, noise, occlusion and minor changes in viewpoint. But it has the shortcoming of high computation complexity.

The second category is based on geometrical characteristics. These methods operate by examining each pixel in the context of its neighbourhood to see if it the pattern formed looks like a corner. Our proposed corner detector algorithm belongs to this category.

Smith and Brady [7] introduced the SUSAN corner detector as follows: Consider any arbitrary pixel in the image (called the 'nucleus') and corresponding circular mask around it. An area of the mask where pixels have the same or similar brightness as the nucleus is defined as "USAN", an acronym standing for "Uni-value Segment Assimilating Nucleus". A pixel is detected to be a candidate corner when the USAN area around it is a local minimum. Trajkovic and Hedl [8] use a similar idea by examining the pixel values at either end of a diameter line across the discredited circle. The response function is defined as

$$C = \min(f_p - f_c)^2 + (f_p - f_c)^2 . \text{ Equation 1}$$

Where f_c is the pixel value at the centre of the circle, and f_p and f_p are the pixel values at either end of a diameter line across the circle. The value of C can only be large in the case of the corner. The Trajkovic corner detection algorithm has improved computational efficiency. However, it is sensitive to noise along edges and at corners, and is overly sensitive to diagonal edges.

The fastest corner detection algorithm, is simply known as, FAST, and uses machine learning methods to classify pixels as corners or non-corners. In [9], a three-layer neural network is trained to recognise corners

where edges meet at a multiple of 45 degree, near to the centre of an 8x8 window. It classifies a pixel p , forming the centre of a circle of 16 pixels as a corner if there are a set of n contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel, p , plus a threshold t , or all darker than p . If any three or more of the four connected neighbouring pixels are all brighter or darker, p is classified as the corner. FAST then employs decision tree ID3 classification to yield more information about whether the candidate pixel is a corner. FAST is however very sensitive to noise and is unstable as shown by results described in section 3.

2. Pattern-based corner detection algorithm

In this section we introduce our new corner detection algorithm. We also compare its performance with other popular detectors using the following criteria:

A. Evaluation Criteria

To be useful for feature point matching, a corner detector should satisfy the following criteria:

- All "true corners" should be detected.
- Corner points should be well localized.
- Detector should have a high repeatability rate (good stability) including rotation invariance.
- Detector should be robust with respect to signal noise.
- Detector should be computationally efficient

We shall use these criteria in experimentally evaluating the performance of various corner detection methods that will be discussed in section 3. We note that the first two measures need ground truth data.

B. Algorithm

Considering an 8 connected pixel neighbourhood around the corner candidate p , corners are detected if the adaptively thresholded neighbourhood matches any one of a set of pre-defined binary patterns. There are actually two sets of binary patterns; one defining lines and the other corners. A partial set of valid binary corner patterns are shown in the following diagram (Figure 1, a, b, c, d). The full set of patterns is created by rotating each of these four patterns through angles of 90, 180, 270 degrees.

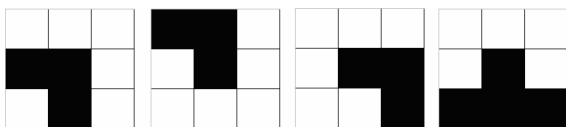


Figure 1 a, b, c, d

For example, corner pattern in figure 1.d can have four associated rotations in figure 2.

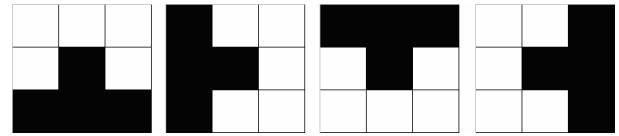


Figure 2 a, b, c, d

To reduce computational complexity, the algorithm first tests each candidate point to see if its neighbourhood conforms to the patterns that define lines, which are listed in Figure 3. This is because there is a smaller set of line patterns to test against and the symmetrical nature of line patterns permits them to be identified more readily than corners. Candidate points found to match against one of these line patterns are rejected and excluded from further analysis..

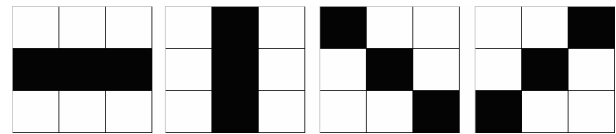


Figure 3 a, b, c, d

Since corners may exist as both luminance minima as well a maxima in a neighbourhood the patterns need to be inverted to detect all possible corners. (i.e. white on black as well as black on white). Yet rather than invert all the sixteen positive corner patterns themselves to make thirty two positive and negative patterns that each candidate pixel needs to be checked against we instead invert the input image and check it twice against one set of positive patterns.

1) Noise Reduction: Most corner detectors use smoothing filters to reduce high frequency noise in images and thereby reduce their noise sensitivity. Both Harris and Trajkovic pre-process the images by using a Gaussian filter to remove the noise. Mokhtarian, Baumberg, and Lindeberg [10,11,12] use multi-scale smoothing where the smoothing filter is applied to the image at different resolution levels and then corner detection is applied over those smoothed images. Smoothing however generates information loss, such as spatial aliasing that reduces the ability to accurately localise corners, and it may also suppress corners altogether by filtering them out. In contrast to the spatial information loss obtained with the use of smoothing filters to eliminate signal noise our approach replaces the random signal noise with controlled (adaptive) quantisation noise. This preserves the spatial information of all detected corners accurately, although some corners may still go undetected due to the quantisation noise.

2) Invariant to Luminance Change: Since the thresholds used in the binarisation mechanism are adapted based on local image statistics the algorithm is invariant to both global and local luminance variations.

3) Corner Response Function: The corner response function operates by considering a 3x3 neighbourhood

where the candidate corner p is in the centre. The corner detector classifies p as a corner if the candidate pixel p satisfies three criteria. First, the variation of pixels, v_p , within its local 3x3 neighbourhood is greater than the variation, v_r , in its general background region r , where the background region r is defined as a 32x32 window.

$$v_p > v_r,$$

where

$$Var_p = (\sum_{i=0}^3 x_i) / 4 - (\sum_{i=5}^8 x_i) / 4 \quad \text{Equation 2}$$

$$Mean_p = (\sum_{i=0}^3 x_i + \sum_{i=5}^8 x_i) / 8 \quad \text{Equation 3}$$

where X_i is sorted elements in local 3x3 region.

$$V_r = \begin{cases} c * \sigma_r \\ t \end{cases} \quad \text{Equation 4}$$

where σ is standard deviation in background region r times a constant c , c is between 0 to 1, and t is predefined threshold between from 0 to 255.

Secondly, each region matches one of the predefined binary patterns. For each pattern listed in figure 1, there is a set of rotated and translated patterns need to be included as well. Thirdly the number of differences, d , is between one and three, where d , is defined as the sum of logic XOR elements on two binary elements crossing centre element X_4 , where x_0, \dots, x_8 are elements in each 3x3 patch arranged in the order from left to right, top to bottom.

$$d = x_0 \oplus x_8 + x_2 \oplus x_6 + x_1 \oplus x_7 + x_3 \oplus x_5 \quad \text{Equation 5}$$

4) False and Adjacent Suppression: There may be some regions, generated by the noises that represent isolated points as shown in Figure 4. To remove these isolated points, we check the number of differences found at each binary patch, most of those pixels have zero differences and consequently discarded.

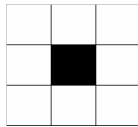


Figure 4

The proposed corner algorithm is described as follows:

For each candidate pixel P in the image, a 3x3 local region is extracted.

If there no adjacent corner to pixel P ,

Calculate the intensity variation VAR_p and $Mean_p$ within the local 3x3 region using equations 3 and 4 respectively.

If the variation within the local region is less than the minimum threshold, skip.

Otherwise, threshold the pixels in the region to turn them either ON or OFF. Each pixel X_i with intensity value less than local threshold, $Mean_p$ in the region, is

set to OFF, while pixels with intensity higher than local threshold are set to ON status.

$$X_i = \begin{cases} 0 & , X_i \leq Mean_p \\ 1 & , X_i > Mean_p \end{cases} \quad \text{Equation 6}$$

If the status of centre pixel is OFF, negate every element in local region to generate the negative complement of the region and retest this pattern against the set of defined patterns.

Remove the line patterns first.

If the number of difference is within the range, store the pixel P as corner.

3. Experiments and results

In this section, we give some experimental results obtained by running Harris, FAST, SIFT and our corner detectors on natural indoor images (Figure 5.a).

To compare noise robustness we consider the statistical performance of different corner detectors under varying conditions. First we evaluate their performance under different amounts of speckle noise (0%, 1%, 2%,) and determine the number of corner points detected at each level using for our test data the images in figure 5 and 10. From the results of the number of corners detected as shown in table 1 and figure 6, 7, 11 and 12, we can observe that the computationally intensive SIFT detector has the best noise robustness under varying amounts of speckle noise as would be expected.. Our algorithm which is five times faster (see table 4) is almost as good as SIFT and is markedly better than all the other algorithms tested.

Table 1 Comparisons of corners detected with Speckle noise

Speckle noise %	FAST	Harris	SIFT	Pattern-Based
Office				
0	2591	1421	437	419
1	4944	2250	499	481
2	7088	3350	522	580
Variation	174%	136%	19%	38%
Hallway				
0	1981	1008	597	419
1	5444	4925	753	580
2	8997	9410	879	744
Variation	354%	834%	47%	78%

The low pass Gaussian filtering performed by existing corner detection methods has the effect of removing any random signal noise in the images. Hence robustness to white noise is fairly similar in all these methods and directly dependent on the amount of filtering performed. For this reason no comparison is presented for white noise performance.

Corner detection used for calculating optical flow or motion often needs to handle cases where the input

image is undergoing rotation. In this case rotational invariance is important. In order to compare the rotation invariance, we have rotated the test image by 0°, 3°, 10°, and 12°, before applying the different corner detection algorithms. This range of rotation angles is typical of applications in automatic video object registration, segmentation and object tracking. Due to the high frame rates in video it is unlikely to have large rotation angles between frames since the rotation would have to be extremely fast (about 60 rpm) to generate higher angles within the 0.04 second interval between frames. The results of these experiments are shown in table 2, figure 8 and 13. We can observe that our proposed corner detector is more rotation invariant than all the other algorithms tested including the SIFT algorithm because it consistently returns close to the same number of corners irrespective of rotation angle.

Table 2 Comparisons of corners detected with rotations

Rotation in Degree	FAST	Harris	SIFT	Pattern-Based
Office				
0°	2591	1421	437	419
3°	2592	1491	480	449
10°	3146	1502	493	413
12°	2907	1659	498	447
Variation	21%	17%	14%	9%
Hallway				
0°	1981	1008	597	419
3°	1651	755	484	459
10°	6705	1556	720	483
12°	1698	732	492	405
Variation	306%	113%	49%	19%

We also have compared detection performance under luminance variations within a 40% change in table 3, figure 9 and 14. Most of the algorithms are fairly robust in this area with very similar performance of around a 2% to 4% variation in the number of corners detected. The FAST algorithm stands out as the worst performer by a significant margin.

Table 3 Comparisons of corners detected with luminance

Luminance Change	FAST	Harris	SIFT	Pattern-Based
Office				
-20%	2513	1386	433	404
-10%	2587	1421	438	415
0	2591	1421	437	419
10%	2188	1421	443	412
20%	2511	1381	434	408
Variation	18.4%	2.9%	2.3%	3.7%
Hallway				
-20%	1825	959	548	400
-10%	1968	1006	665	418
0	1981	1008	597	419

10%	1931	1012	605	409
20%	1885	1002	663	410
Variation	9%	6%	21%	5%

The last experiment performed was to evaluate the computation time required by each of the four algorithms. The timing test was performed on a 1.6GHz Pentium 4 processor. From the results shown in table 4 using a 512x512 image of Lena we can observe that the FAST algorithm requires the least computation which is reflected in its otherwise poor performance. Our proposed algorithm is the next fastest and is 4.5 times faster than the SIFT algorithm.

Table 4 Speed comparisons of Corners Detectors

Lena	FAST	Harris	SIFT	Pattern-Based
ms	227	1024	3743	837

4. Conclusion

In this work, corners in images are identified by examining a small patch of an image against a pattern set to see if it 'looks' like a corner. This corner detector is computationally simple since only a small number of patterns are examined for each corner detected. Proposed pattern-based algorithm has achieved better localisation, noise robust, more rotation invariant and invariant to luminance change in comparison to Harris and FAST algorithm. The simplicity of its implementation is also proved and tested with timing experiments.

Acknowledgement

The authors gratefully acknowledge the support of the NICTA Queensland Research Lab in carrying out this research.

References

- [1] M.Y.I. Idris, H. Arof, E.M. Tamil, N.M. Noor and Z. Razak, "Review of Feature Detection Techniques for Simultaneous Localization and Mapping and System on Chip Approach", *Information Technology Journal* 8 (3): 250-262, 2009.
- [2] H. P. Moravec, "Towards Automatic Visual Obstacle Avoidance", *Proc. 5th International Joint Conference on Artificial Intelligence*, pp. 584, 1977.
- [3] C. Harris and M. Stephens, "A Combined Corner and Edge Detector", *Proc. Alvey Vision Conf.*, Univ. Manchester, pp. 147-151, 1988.
- [4] Z.Zheng, H.Wang and Ekteoh, "Analysis of Gray Level Corner Detection", *Pattern Recognition Letters*, Vol. 20, pp. 149-162, 1999.
- [5] W. Forstner and E. Gulch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features", *ISPRS Intercommission Workshop*, Interlaken, pages 149--155, June 1987.

[6] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Key Points", *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004.

[7] S.M. Smith and M. Brady, "SUSAN - A New Approach to Low Level Image Processing", *International Journal of Computer Vision*, Vol. 23(1), pp. 45-78, 1997.

[8] M. Trajkovic and M. Hedley, "Fast Corner Detection. Image and Vision Computing", Vol. 16(2), pp. 75-87, 1998.

[9] Rosten, Edward, Drummond, Tom, "Machine Learning for High-Speed Corner Detection", *ECCV06*, pp. 430-443.

[10] F.Mokhtarian and R.Suomela, "Robust image corner detection through curvature scale space", *IEEE Trans on Pattern Analysis and Machine Intelligence*, 20(12): 1376-1381, 1998.

[11] A.Baumberg, "Reliable feature matching across widely separated views", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1774-1781, 2000.

[12] T.Lindeberg, "Feature detection with automatic scale selection", *International Journal of Computer Vision*, Vol. 30, pp. 77-116. 1998.



Figure 5 office



Figure 6 pattern detection with 0% noise on office



Figure 7 pattern detection with 2% noise on office



Figure 8 pattern detection with 12° rotation on office



Figure 9 pattern detection with -20% luminance on office



Figure 12 pattern detection with 2% noise on hallway

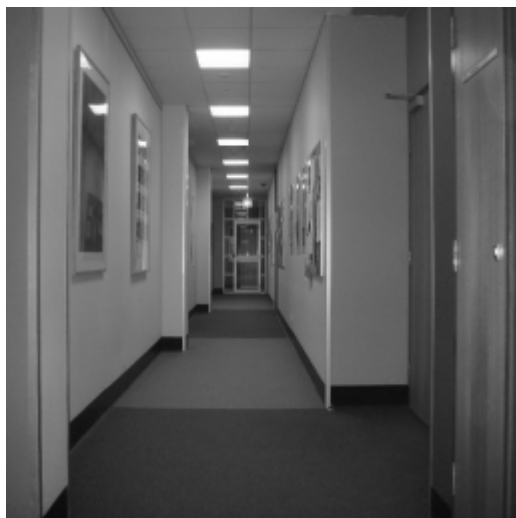


Figure 10 hallway

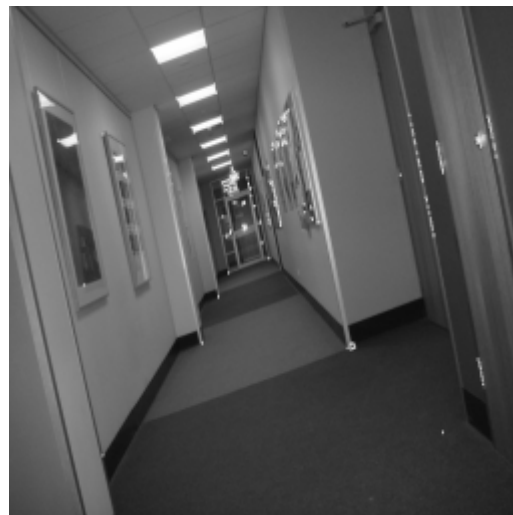


Figure 13 pattern detection with 12° rotation on hallway

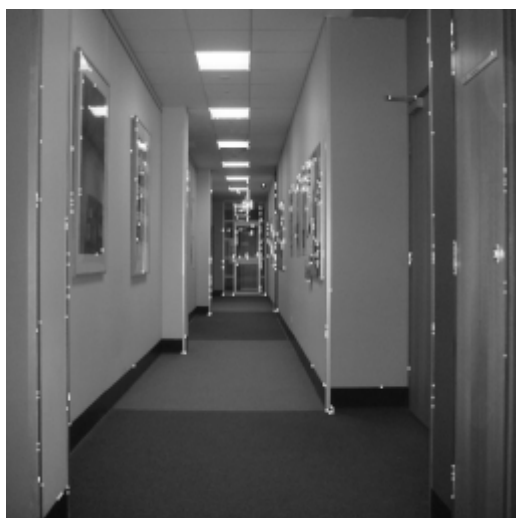


Figure 11 pattern detection with 0% noise on hallway

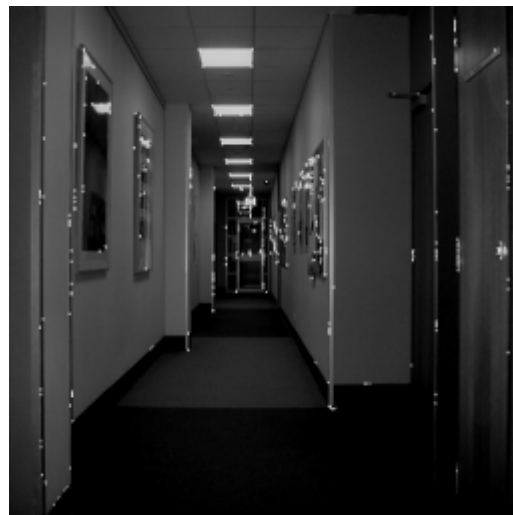


Figure 14 pattern detection with -20% luminance on hallway