

FAST, BIOLOGICALLY INSPIRED CORNER DETECTION USING A SQUARE SPIRAL ADDRESS SCHEME AND ARTIFICIAL EYE TREMOR

John Fegan¹, Sonya Coleman¹, Dermot Kerr¹, Bryan Scotney²

¹School of Computing, Engineering and Intelligent Systems

²School of Computing and Information Engineering

Ulster University, Northern Ireland

ABSTRACT

This paper presents an efficient approach to corner detection in images using a spiral addressing scheme in conjunction with simulated, biological involuntary eye movements. As part of this approach, a combined gradient detection and smoothing operation is used to quickly obtain a feature representation that can be used with a standard ‘cornerness’ measure. A computationally efficient use of the spiral address scheme to apply further processing operations such as non-maximum suppression is demonstrated. Comparative evaluation of three corner detection methods is presented and results demonstrate significantly faster processing times than other well-known corner detection methods.

Index Terms—Fast Image Processing, Square Spiral Address Scheme, Eye Tremor, Corner Detection

1. INTRODUCTION

While fast runtime performance is essential in many machine vision tasks it is difficult to attain using traditional frame-based image processing methods. These methods are based on the assumption that an image uses a raster address scheme where the pixels are stored in computer memory as a matrix and processed in sequential order for each frame. Research indicates that this Traditional Image Processing (TIP) approach does not accurately reflect the way in which biological vision systems process visual information in an efficient manner [1, 2]. Characteristics of the Human Visual System (HVS) have motivated research into hexagonal based image representations where hexagonal based pixels are used with a spiral address scheme [1]. Whilst hexagonal based approaches have demonstrated they are able to achieve increased accuracy and faster runtime performance than TIP [3] they are unfortunately limited in application by a lack of hexagonal pixel image hardware. When a hexagonal image representation needs to be used with standard rectangular pixel-based hardware, the additional processing overhead which is needed to convert a rectangular image to a hexagonal format [3, 4] negates some of the performance improvements. To circumvent this problem, a one-dimensional (1D) square-spiral (squiral) address scheme has been proposed for standard rectangular based images [4] to increase performance. Like hexagonal approaches, Squiral Image Processing (SIP) can achieve faster runtime performance than

TIP [3] and is efficient for low level image processing tasks such as edge detection [3, 5]. Despite these benefits there are some difficulties with using SIP for increasingly complex image processing operations such as corner detection [6]. In this paper we embrace the SIP image framework to develop a computationally efficient corner detection approach.

2. SQUARE SPIRAL IMAGE PROCESSING

Neighbourhood operations are a standard image processing feature extraction technique where each pixel and its surrounding neighbours are processed as a collective unit called a neighbourhood. In TIP this is achieved by iterating over an entire image, pixel-by-pixel, in order to process the complete image. During this operation the adjacent neighbours of one pixel may also be the adjacent neighbours of another nearby pixel, thus the neighbourhoods are said to be overlapping. Biological visual processing systems, in particular the retina, do not possess this overlapping processing ability. In biological vision systems the cells that sense and process visual stimuli are arranged in collective groups called receptive fields (RFs) [2, 7]. The visual stimuli sensed through photoreceptors and processed by intermediate retinal cells, which are arranged in RFs, correspond with a lesser number of retinal ganglion cell outputs which indicates that early visual processing in the HVS is not coordinated by substantial overlap of RFs [2].

The squiral address scheme simulates the arrangement of RFs by grouping pixels into sets of non-overlapping but contiguous neighbourhoods (Figure 1) where pixels in each neighbourhood are indexed consecutively. To achieve this an image is partitioned into square clusters called layers (denoted as λ). The address scheme begins at layer 0, which is the origin pixel at the centre of an image and spirals over the eight surrounding adjacent neighbours to form the 3×3 layer 1. Thereafter the eight neighbourhoods that surround layer 1 are addressed recursively in the same spiral order to form layer 2 which is composed of eight 3×3 neighbourhoods. Likewise, layer 3 encompasses layer 2 and eight surrounding neighbourhoods, recursively addressed in the same way. This partitioning approach is repeated for all subsequent layers when using the squiral address scheme [4]. The scheme naturally facilitates neighbourhood processing operations because each pixel at a $0 \bmod 9$ address (the centre of a squiral neighbourhood) can be processed with its

neighbours in a contiguous sequence. However, in the case of pixels which are not at the centre of a squiral neighbourhood, computationally expensive mod 9 addition and multiplication operations are required to find their neighbours. Hence, the squiral address scheme is biologically inspired in that it embraces non-overlapping operations, where only pixels at the centre of a squiral neighbourhood are processed. This approach is efficient but results in a sparse or downsampled image output which on its own is not always sufficient for computer vision tasks. In Section 2.1, an approach inspired by biological eye tremor is outlined to obtain non-sparse outputs from non-overlapping operations.

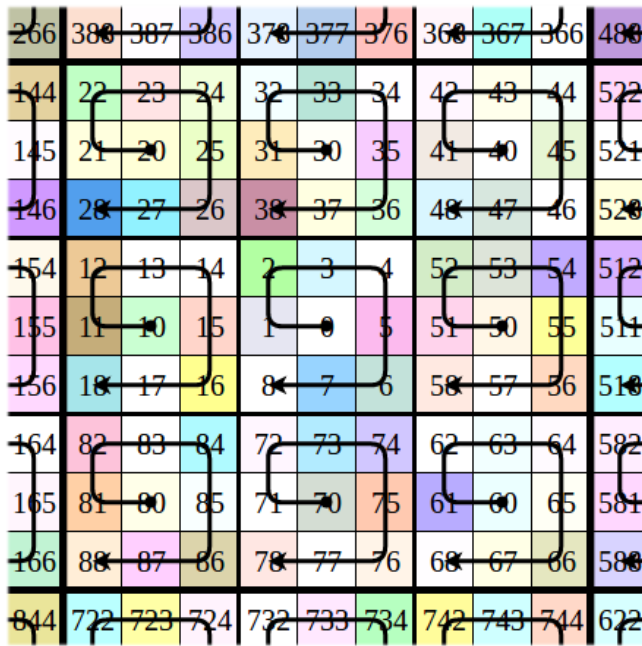


Figure 1. Square Spiral Address Scheme

2.1. Eye Tremor

Biological eye tremor is thought to maintain activity in the retina, leading to visual perception [8]. Taking inspiration from this biological process we can simulate eye tremor by capturing a number of image frames at slightly different spatial locations where each image corresponds to visual perception from a shifted eye movement. In SIP eye tremor is simulated by shifting the origin of the squiral address scheme over each initial layer one address. In practice the address scheme shifts each time a new frame is captured, so the centre of each squiral neighbourhood is focused on a different mod 9 neighbour location. Thus, each of the pixels at each mod 9 address can be sparsely processed over nine image frames and their outputs are subsequently combined to generate a non-sparse output. This is demonstrated in Figure 2 where nine squiral, eye tremor frames (denoted as $F_0 - F_8$) are obtained using the image in Figure 1 and represented as 1D arrays. For each of these frames, each squiral

neighbourhood is processed and the pixel at the centre is used to construct as part of an output (denoted as O in Figure 2). Eye tremor is applied here to the same image with the origin shifted over each layer one address, this results in 9 eye-tremor images (the original image and eight shifted images) which are processed to obtain a non-sparse output.

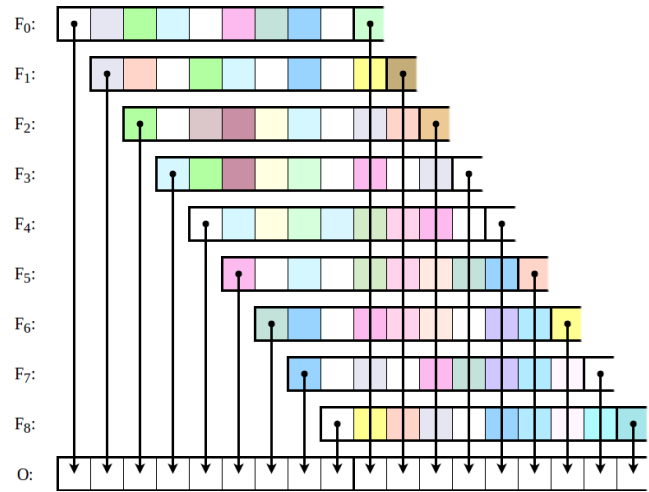


Figure 2. Eye Tremor Processing

3. SIP CORNER DETECTION

A common approach to corner detection is based on an image gradient: changes in luminance between pixels and their neighbours [9]. A gradient is said to be large where there are prominent changes and small where there are few or no changes. In other words, a gradient is a representation of discontinuities in an image making it useful in tasks such as feature detection. For example, an edge is perceived as a prominent and consistent discontinuity in luminance; therefore, edges can occur where an image gradient is large in a single direction. By extension, corners are usually locations where two or more edges intersect, more precisely, a location where an image gradient is large in multiple directions. In practice, gradient based corner detection can have several stages, but the three common ones are: gradient detection; gradient smoothing and corner measurement; and non-maximum suppression.

3.1. Gradient Detection

A common way to measure an image gradient involves convolution: the multiplication of a matrix operator on a pixel and its neighbours followed by the summation of their products. In TIP, convolution normally involves sliding a matrix operator over an image, pixel-by-pixel, such that the neighbourhoods covered by the operator overlap. In SIP, operations involving overlapping neighbourhoods are inefficient and do not align with the way biological RFs are arranged. The use of eye tremor in conjunction with SIP

permits sparse gradient detection where the gradient detection operator is vectorised using the squiral address scheme and applied contiguously to each pixel at the centre of a squiral neighbourhood. The non-overlapping SIP approach is fast and relatively straightforward to implement as demonstrated in [3] and [5] where SIP is used with squiral gradient detection operators to find edges in images and videos.

3.2. Gradient Smoothing and Corner Measurement

Often in gradient based corner detection the components of an image gradient must be smoothed to reduce noise: inherent variance between pixels that can cause inaccurate feature detection [9]. Smoothing is usually achieved using convolution with an operator that reduces the variance between pixels and their neighbours. In TIP, smoothing an image gradient is straightforward because operations can have overlapping neighbourhoods and are therefore able to produce non-sparse outputs. However, in SIP smoothing an image gradient is problematic because the image gradient is stored in a squiral vector (O in Figure 2) which does not easily permit overlapping operations. In this situation, the addresses of a pixel's neighbours can be found using computationally expensive base 9 arithmetic. To avoid this problem an operator that concurrently detects and smooths the gradient of an image can be used; in this case, the 3×3 Linear² Gaussian (L^2G) operator [10] which measures a product of the first order directional derivative:

$$H_l^\sigma(U) = \int_{\Omega} (\underline{b}_l^1 \cdot \nabla U) (\underline{b}_l^2 \cdot \nabla U) \xi_l^\sigma d\Omega \quad (1)$$

Here, two piecewise linear basis functions are used with a Gaussian basis function to generate the non-linear operators $H_{I_x I_y}^\sigma$, $H_{I_x^2}^\sigma$ and $H_{I_y^2}^\sigma$. In the case of $H_{I_x I_y}^\sigma$, \underline{b}_l^1 and \underline{b}_l^2 are along the x and y coordinate directions, respectively; and for $H_{I_x^2}^\sigma$ and $H_{I_y^2}^\sigma$, $\underline{b}_l^1 = \underline{b}_l^2$ along the x and y coordinate directions. Hence, a direct measure of the products of directional derivatives is obtained, rather than the products from measures of the directional derivatives. In-built smoothing is performed via the presence of a Gaussian basis function in the operator definition. The corner strength response can then be calculated as in [9], but now using the cornerness measure:

$$C = (H_{I_x^2}^\sigma H_{I_y^2}^\sigma - (H_{I_x I_y}^\sigma)^2) - k(H_{I_x^2}^\sigma + H_{I_y^2}^\sigma)^2 \quad (2)$$

3.3. Non-Maximum Suppression

The final step in corner detection is Non-Maximum Suppression (NMS): an operation which suppresses non-maximum values in a pixel neighbourhood. Incidentally, NMS can incorporate thresholding to suppress maximum values that are not within a certain range. In corner detection, NMS is used to select strongly responding corner points. The

most common NMS method in TIP [11] is presented in Algorithm 1: here the pixels of an image are compared with their neighbours to determine if they are neighbourhood maximums. In this algorithm: r is the radius of the neighbourhood, u and v are coordinates used to navigate the neighbourhood; p and q are the upper boundaries of the neighbourhood; I is a raster input matrix; and O is a raster output matrix.

```

u = x - r, p = x + r
while (u ≤ p)
    v = y - r, q = y + r
    while (v ≤ q)
        if (Ix,y < Iu,v) {goto end}
        v = v + 1
    u = u + 1
Ox,y = Ix,y
end: y = y + 1

```

Algorithm 1: Traditional Non-Maximum Suppression

In Algorithm 2 a non-overlapping SIP based NMS suppression method is presented: here a single maximum value is retained from each squiral neighbourhood. In this algorithm: u is the coordinate used to navigate the neighbourhood; a is the size of the neighbourhood; p is the upper boundary of the neighbourhood; \max is the address of the pixel with the largest value; I is a squiral input vector and O is a squiral output vector.

```

u = x + 1, p = x + a, max = x
while (u < p)
    if (Iu > Imax) {max = u}
    u = u + 1
Omax = Imax
x = p

```

Algorithm 2: Squiral Non-Maximum Suppression

It can be seen that the squiral NMS algorithm has a more streamlined appearance than its traditional counterpart. This is reflected in the computation which is significantly less in the squiral algorithm. More precisely, the traditional algorithm can require up to $p \times q$ comparisons per pixel, whereas the squiral algorithm requires only one comparison per pixel regardless of the neighbourhood size. This is achieved via non-overlapping neighbourhood suppression and becomes increasingly important as the size of the NMS neighbourhood grows. For example, as the size of p and q increase the traditional algorithm requires more comparisons. In the squiral algorithm p can be scaled without any additional computational overhead.

4. PERFORMANCE EVALUATION

The efficiency of corner detection is evaluated using three approaches. First, a 'Harris' approach using TIP, a single input image and three operators: the 3×3 Prewitt operator [12] for gradient detection; a 3×3 Gaussian operator with a

standard deviation of 0.85 for gradient smoothing; and the NMS method in Algorithm 1 - this can be considered analogous to the well-known Harris corner detector [9]. Second, an ‘L²G’ approach using TIP, a single input image and two operators: the L²G operator for concurrent gradient detection and smoothing; and the NMS method in Algorithm 1. Lastly, our ‘Bioinspired’ approach using SIP with an eye tremor image set, the L²G combined edge and corner detector [10] and the NMS method in Algorithm 2.

4.1. Runtime Evaluation

Table 1 shows the runtimes for the three approaches where different sizes of NMS neighbourhoods (noted in the left most column) are considered. The approaches were implemented in C++ 11 and compiled with GNU g++ version 5.4.0 with default optimisation. The runtimes were measured using the system clock and the approach in [13]. Times are given in seconds and show the time taken to process a 2187 × 2187 image. The runtimes account for the entirety of the corner detection process and are averaged over one hundred execution cycles. In these experiments a system with an Intel Core i7-4790 CPU @ 3.60GHz × 8, 16GB RAM and Ubuntu Linux 16.04 LTS 64-bit was used.

	Harris	L ² G	Bioinspired
3 × 3	0.555724s	0.278108s	0.332516s
9 × 9	1.109690s	0.828181s	0.338572s
27 × 27	6.443590s	5.981650s	0.336046s
81 × 81	59.49180s	54.38980s	0.335076s

Table 1: Corner Detection Runtimes

The timings show that the L²G approach is faster than the Harris approach in all cases. This is more notable if you consider the overhead of NMS. This supports the findings in [10] and shows that gradient detection and smoothing are faster when computed in combination rather than separately. When a 3 × 3 NMS neighbourhood is considered, the L²G approach is faster than the Bioinspired approach, though this could be explained by the overhead needed to switch between nine eye tremor images. Incidentally, the runtimes for the Bioinspired approach are the sums of the times needed to process nine frames. Thus, the time needed to process a single squiral frame is approximately one-ninth of the times given (0.036951s when measured using the system clock) and a substantial improvement over TIP. It can also be seen that as the size of a NMS neighbourhood grows, the runtimes for the Harris and L²G approaches increase exponentially. However, the runtimes for the Bioinspired approach remain constant showing that this approach has significant potential for applications where large-scale NMS is necessary.

4.2. Visual Evaluation

A visual comparison of the corner maps obtained from each approach are presented in Figure 3 and demonstrate that improved run-time performance does not significantly reduce accuracy. The corner maps were obtained using a 27 × 27 NMS neighbourhood with thresholding to achieve results that offer visual best representations. It should be noted that several images have been used to test each program and their results are in line with the quality of those presented here.

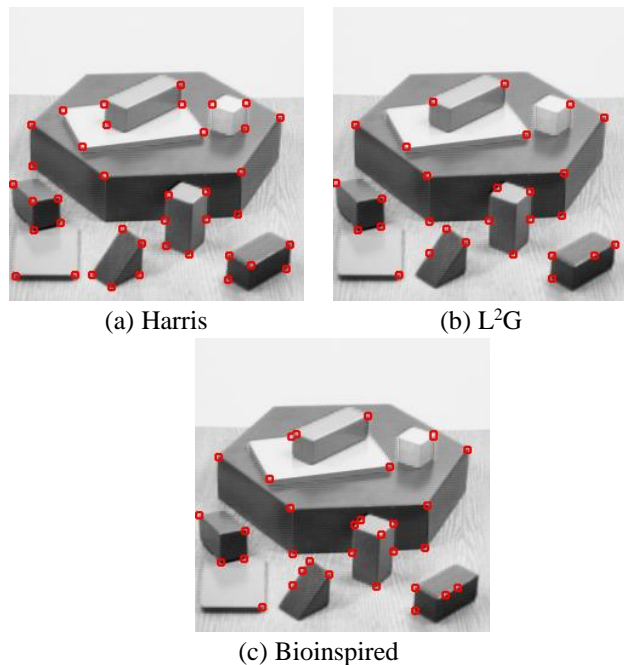


Figure 3. Corner Maps

The visual results show that the Harris approach detected the most locationally accurate corners and a greater number of corners than the two other approaches which use the L²G operator. This is a concession of the L²G operator which offers increased computational performance but reduced detection performance [9]. More importantly, the SIP approach detected almost all the corners found in the L²G approach and some additional ‘false’ corners near their vicinity. The detection of these additional corners can be explained by the squiral NMS method which can occasionally retain corners at the borders of two adjoining squiral neighbourhoods. Nonetheless, most of the corners detected are accurate showing that this approach can be used in corner detection tasks and ultimately in real-time robot vision tasks such as navigation or tracking.

5. CONCLUSION

In this paper, a fast and accurate approach to corner detection is presented which is based on adapting the traditional principles of gradient based corner detection for use with a

1D spiral address scheme using a bioinspired eye tremor framework. The results of the corner detection experimentation show that this approach is slightly less accurate than TIP methods, but also significantly faster, especially when large scale NMS is needed. Further work will seek to implement invariant feature detectors and extend the developed SIP framework to other topics within computer and robot vision such as object recognition, high speed navigation and tracking.

6. REFERENCES

[1] Middleton, L., Sivaswamy, J., “Hexagonal Image Processing - A Practical Approach”, Springer, 2005.

[2] Róka, A., Csapó, Á., Reskó, B., Baranyi, P., “Edge Detection Model Based on Involuntary Eye Movements of the Eye-Retina System”, *Acta Polytechnia Hungarica*, vol. 4, no. 1, pp. 31 - 46, 2007.

[3] Jing, M., Scotney, B., Coleman S., McGinnity, M., “Novel ‘Squirrel’ (Square Spiral) Architecture for Fast Image Processing”, *Journal of Visual Communication and Image Representation*, pp. 371 - 381, 2017.

[4] Jing, M., Scotney, B., Coleman S., McGinnity, M., “A Novel Spiral Addressing Scheme for Rectangular Images”, *International Conference on Machine Vision Applications*, pp. 102 - 105, 2015.

[5] Fegan, J., Coleman, S., Kerr, D., Scotney B., Fast Video Processing Using a Spiral Coordinate System and an Eye Tremor Sampling Scheme, *Irish Machine Vision and Image Processing Conference Proceedings*, 2017.

[6] Fegan, J., Coleman, S., Kerr, D., Scotney B., “Fast Corner Detection Using a Spiral Architecture”, *Irish Machine Vision and Image Processing Conference Proceedings*, pp. 51 - 57, 2016.

[7] Vance, P., Das, G., Kerr, D., Coleman, S., McGinnity, M., Gollisch, T., Liu, J., “Bioinspired Approach to Modeling Retinal Ganglion Cells Using System Identification Techniques”, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1 - 13, 2017.

[8] Martinez-Conde, S., Macknik, S. L., Hubel, D. H., “The Role of Fixational Eye Movements in Visual Perception” *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229-240, 2014.

[9] Harris, C., Stephens, M., “A Combined Corner and Edge Detector”, *Proceeding of the 4th Alvey Vision Conference*, pp. 147 - 151, 1988.

[10] Coleman, S., Kerr, D., Scotney B., “Concurrent Edge and Corner Detection”, *IEEE International Conference on Image Processing*, pp. 273 - 276, 2007.

[11] Pham, T., “Non-maximum Suppression Using Fewer than Two Comparisons per Pixel”, *Advanced Concepts for Intelligent Vision Systems*, pp. 438 - 451, 2010.

[12] Judith, M., Prewitt, S., “Object Enhancement and Extraction”, *Picture Processing and Psychopictorics*, pp. 75 - 149, 1970.

[13] Cplusplus.com, “Clock - C++ Reference”, Available: <http://www.cplusplus.com/reference/ctime/clock/>. [Accessed: 01-Feb-2018]