

An Implementation Framework for Fast Image Processing

J. Fegan, S. A. Coleman, D. Kerr
School of Computing and Intelligent Systems,
University of Ulster, Magee College,
Londonderry, N. Ireland, U.K.
e-mail: {fegan-J2, d.kerr, sa.coleman}@ulster.ac.uk

B. W. Scotney
School of Computing and Information Engineering, University of Ulster, Coleraine,
N. Ireland, UK

© 20xx IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This publication appears in:

2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS 2017)

2017 3rd International Conference on Robotics and Vision (ICRV 2017)

June 16-18, 2017

Wuhan, China.

An Implementation Framework for Fast Image Processing

J. Fegan, S. A. Coleman, D. Kerr

School of Computing and Intelligent Systems,
University of Ulster, Magee College,
Londonderry, N. Ireland, U.K.

e-mail: {fegan-J2, d.kerr, sa.coleman}@ulster.ac.uk

B. W. Scotney

School of Computing and Information Engineering,
University of Ulster, Coleraine,
N. Ireland, UK

e-mail: bw.scotney@ulster.ac.uk

Abstract—Efficient processing of image data is a key aspect of achieving real-time performance for image and video applications. Here, a biologically inspired novel framework which uses a spiral indexing scheme is used to facilitate fast image processing. In particular we demonstrate the effectiveness of our approach on low-level image operations (convolution) and feature extraction (edge detection). Unlike conventional image addressing schemes where the pixels are indexed using two-dimensional Cartesian coordinates, a spiral addressing scheme enables the pixels to be stored in memory adjacent to their immediate neighbours and indexed as a one-dimensional vector. This permits both efficient traversal of the image structure and efficient application of image processing operators. Performance is evaluated by the application of Laplacian edge detection. The results demonstrate the efficiency of the proposed approach compared with a typical two-dimensional implementation

Keywords—feature extraction; image frameworks; squirrel image

I. INTRODUCTION

In recent years research has embraced the idea that biological processes can be thought of as algorithms which may be subsequently emulated in computer hardware and used to help solve computational problems. Many areas of computer science have benefited from this knowledge of biological systems resulting in developments such as genetic algorithms [1], neural networks [2], and deep-learning [3]. In terms of computer vision, one of the early pioneers was David Marr [4] who integrated results from psychology, artificial intelligence, and neurophysiology into new models of visual processing. Even so, the basic problem of fast and reliable detection, recognition and categorisation of visual scenes is an area of active research.

Whilst considerable progress has been made in understanding how biology can help solve computer vision problems this has been largely based upon the intuitive notion that two-dimensional (2D) visual data can be sampled as a matrix of picture elements (pixels) using a rectangular lattice of sensors with a Cartesian coordinate system [5]. Recent research has demonstrated that hexagonal pixel-based images which closely emulate the spatial arrangement of photoreceptors within the retina offer advantages in terms of image capture and analysis. Building on this approach Sheridan [6] introduced a novel spiral addressing scheme in which each hexagonal pixel can be indexed using a single co-ordinate address, rather than the typical 2D cartesian address that is used with regular rectangular digital images.

Whilst such a one-dimensional addressing scheme potentially provides an appropriate structure for real-time image processing of hexagonal images the computational advance of hexagonal based approaches is hindered by the additional time and effort required for conversion of image data to a hexagonal environment, as existing hardware for image capture and display are based predominantly on traditional square pixels. In [7] a novel “squirrel” (square spiral) image processing framework is proposed which uses an efficient spiral addressing scheme for standard square images. Unlike the hexagonal based approaches, “squirrel” image processing can be achieved easily using an existing lattice with a Cartesian coordinate system. In addition this framework naturally embraces parallel computation based on the biological “eye tremor” phenomenon of the human visual system [8] whereby a non-overlapping convolution technique may be used to facilitate fast computation.

In this paper, we present our recent work using the “squirrel” framework with eye tremor and convolution of non-overlapping Laplacian masks which Marr proposed as analogous to the standard on-off receptive fields found in biological vision systems [9]. We also demonstrate how this approach can be expanded for more complex image operations, namely corner detection [10] where we evaluate the performance and effectiveness of this approach. In Section II we outline the “squirrel” architecture and in Section III we present performance evaluation, with the work being concluded in Section IV.

II. THE “SQUIRAL” ARCHITECTURE

A. “Squirrel” Addressing Scheme

Inspired by the work in [7] we use a squirrel addressing scheme for standard square images as shown in Fig. 1. It can be seen in Fig. 1 that the addressing scheme starts at the centre pixel (layer-0), then moves to pixel 1 around to pixel 8 indexed in a clockwise direction. The cluster of the centre pixel together with its eight (layer-0) neighbours is considered as layer 1. Layer 2 is generated by recursive use of layer-1 clusters, such that eight layer-1 clusters are combined to form layer-2. Ultimately, the entire image can be considered as a layer- λ cluster comprising 9^λ pixels. This structure facilitates the use of base 9 numbering to address each pixel within the image using one-dimensional indexing. For example, the pixels in layer-1 are labelled from 0 to 8, indexed in a clockwise direction. The base 9 indexing continues into each layer, e.g. layer-2 starts from 10, 11,

12... and finishes at 88. Subsequent layers are structured recursively. Consequently, the converted image is stored in a one-dimensional vector according to the one-dimensional addresses.

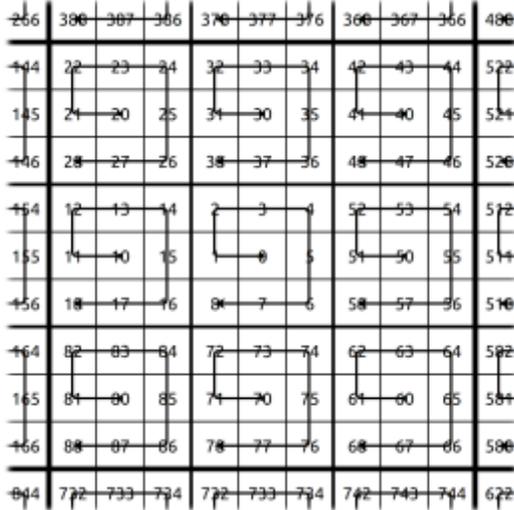


Figure 1. Squiral address scheme

Currently images are captured as 2D images, using a standard Cartesian coordinate scheme, therefore we must convert the images to this new squiral architecture to demonstrate the benefits of such an image structure. As presented in [7], there are three key steps to this: 1) *Determine the size of the squiral image*: The size of a squiral image can be found by the number of layers λ . For a given rectangular image with size $M \times N$, the number of layers λ can be obtained by $\lambda = \log MN / \log 9$, therefore the length of the image is 3^λ . 2) *Squiral Addressing*: a squiral mapping begins at the centre of a square image and spirals outwards using a one-dimensional index sequence. The address can be defined as: $a_{n-1} \dots a_1$, where $0 \leq a_i < 9$ (as the address scheme is base 9). Analogous to the spiral addressing scheme used for hexagonal image processing [6], we can represent the address as:

$$a_{n-1} \dots a_1 = \sum a_i \times 10^{i-1}$$

where Σ denotes Squiral Addition and \times indicates Squiral Multiplication. The Squiral Addition corresponds to translation of the image and the Squiral Multiplication corresponds to rotation and scaling of the image.

3) *Squiral address Location*: For a standard image represented by Cartesian coordinates (x, y) , we define the centre point as $L(0) = (0, 0)$. Based on the Squiral addressing scheme, we can find the points in layer-1: $L(1) = (-1, 0)$, $L(2) = (-1, 1)$, $L(3) = (0, 1)$, $L(4) = (1, 1)$, $L(5) = (1, 0)$, $L(6) = (1, -1)$, $L(7) = (0, -1)$ and $L(8) = (-1, -1)$. To locate the points in a higher layer, we calculate the number of pixels required to shift the point from the centre to the target point by

$$L(a_i \times 10^{i-1}) = 3^{i-1} \times L(a_i)$$

Further details on the Squiral address scheme can be found in [7].

B. Traditional Image Representation

There are two typical ways to arrange an image in computer memory. The first way is to arrange the rows or columns of the image as contiguous blocks (Figure 1). This allows each element to be accessed with an index calculated as:

$$my + x \text{ or } nx + y$$

where m is the image width, n is the image height, y is a vertical image coordinate and x is a horizontal image coordinate. This arrangement offers good locality but in neighbourhood processing its efficiency is challenged by the distance between a pixel and its immediate neighbours. The distance between a pixel and its neighbours, coupled with the excessive multiplication and addition needed to locate those neighbours limits the effectiveness of this arrangement for neighbourhood operations. The second way to arrange a matrix in memory is to store each row or column in separate blocks and use an array of references that point to each block. This approach avoids the computation of the first approach but the rows or columns are no longer contiguous in memory. This hinders memory access and limits runtime performance.

C. Eye Tremor Image Representation

Research on the squiral image processing framework has demonstrated that image processing can be hastened if the pixels of an image are stored adjacent to their immediate neighbours in memory. The layered structure of the squiral address scheme encourages a form of non-overlapping processing where pixel clusters can be processed efficiently by treating values which are stored adjacent to one another. In order to process pixel values that cross the boundaries of more than one layer we use a technique that is inspired by involuntary eye movements called tremors [8]. Eye Tremor is simulated by use of a set of images whereby the pixel values in the original image are shifted 1 or more pixels in either the x , y , or both x and y directions. Here, the original squiral image uses an indexing scheme based on a base-9 layer we construct 8 1-pixel shifted versions of the original image corresponding to the cardinal and intermediate directions. Each of these additional images is shifted spatially from the original image centre by a distance of one pixel. This results in the original image and a corresponding set of eight shifted images as illustrated in Figure 2.

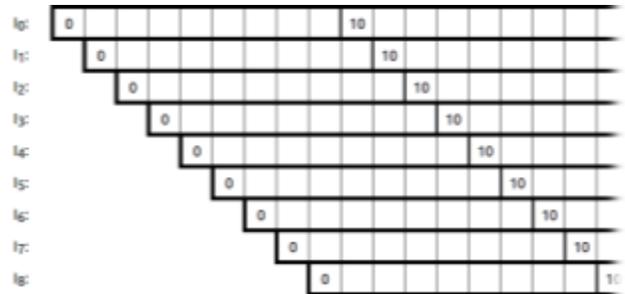


Figure 2. Nine eye tremor images

The eye tremor representation in Figure 2 has been used in [7, 11] and when non-overlapping convolution is applied in the horizontal direction along each 1D image, this results in a set of feature maps ‘*a trous*’ which can then be merged into one feature map. However, a simple but effective improvement on the approach in [11] is to not represent the offset images in several off-set arrays, but instead to represent them as a 2D image block, as illustrated in Figure 3, and in this way convolution can rapidly and effectively be conducted in the vertical direction resulting in the same gradient output in considerably faster time as illustrated in Section IV.

l ₀ :	0								10
l ₁ :	0								10
l ₂ :	0								10
l ₃ :	0								10
l ₄ :	0								10
l ₅ :	0								10
l ₆ :	0								10
l ₇ :	0								10
l ₈ :	0								10

Figure 3. Eye tremor matrix

III. PERFORMANCE EVALUATION

Preliminary experiments were conducted to assess the runtime performance for different image navigation strategies. The experiments tested the time needed to iterate over a complete image and read and assign a value to each pixel. Incidentally, the assignment of a computed value is included to avoid loop removal through compiler optimisation. In the first test a traditional Cartesian image was navigated using the algorithm in equation 1:

$$\sum_{y=1}^{3^\lambda} \sum_{x=1}^{3^\lambda} I_{y,x} = x + x \quad (1)$$

In the second test a corresponding eye tremor image was navigated using the sparse traversal algorithm outlined in equation 2:

$$\sum_{y=1}^9 \sum_{x=1}^{\frac{9^\lambda}{9}} I_{y,9x} = x + x \quad (2)$$

Equation 3 demonstrates how an eye tremor image can be navigated in a linear fashion:

$$\sum_{x=1}^{9^\lambda} I_{1,x} = x + x \quad (3)$$

Table I shows the average time taken to navigate each image structure, using the above algorithms over 10000 iterations. The results show that linear eye tremor traversal (illustrated in Figure 3) is comparable to Cartesian traversal. However the sparse approach took significantly longer than the Linear approach. It is believed that this is due to the jumps between pixels, whereby the pixels are not traversed as sequenced in memory.

TABLE I. IMAGE TRAVERSAL RUNTIMES

Layer	Framework		
	TIP	EIP(Sparse)	EIP (Linear)
1	0.000193	0.000303	0.000177
2	0.001632	0.001845	0.001568
3	0.012651	0.019594	0.012814
4	0.119783	0.170196	0.116198
5	1.029561	1.609905	1.014792
6	9.241499	19.370751	9.114276
7	83.031241	200.460622	82.363814

However, if a 3×3 Laplacian mask is applied to an image, we clearly see in Table II that it is faster to process an eye tremor image than it is to process a standard 2D image. Furthermore, the Linear approach is faster than the 2D standard image convolution and the 1D sparse convolution approach presented in [7]. These results are encouraging and demonstrate that, although the Linear approach is restricted to convolution of operators of size 3×3 only, a significant speed up can be obtained and hence these approach would be suitable for video processing. A graphical representation of these results is presented in Figure 4, illustrating that as the image size increases the Linear approach becomes increasing faster for convolution than the other two approaches.

TABLE II. IMAGE CONVOLUTION RUNTIMES

Layer	Framework		
	TIP	EIP(Sparse)	EIP(Linear)
1	.000335s	.000286s	.000205s
2	.003030s	.002606s	.001802s
3	.027131s	.021872s	.016444s
4	.236634s	.198595s	.149392s
5	2.155788s	1.818246s	1.340087s
6	19.280412s	16.125603s	14.555113s

7	175.178310s	144.857634s	133.182200s
---	-------------	-------------	-------------

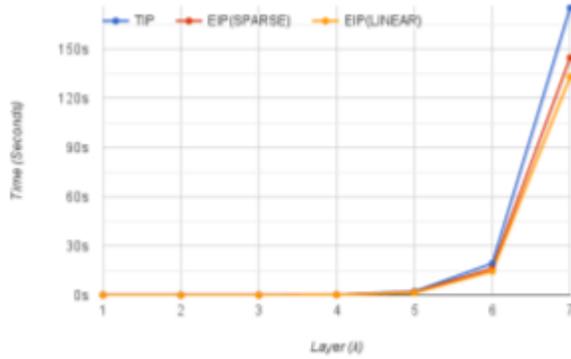


Figure 4. Algorithmic runtimes

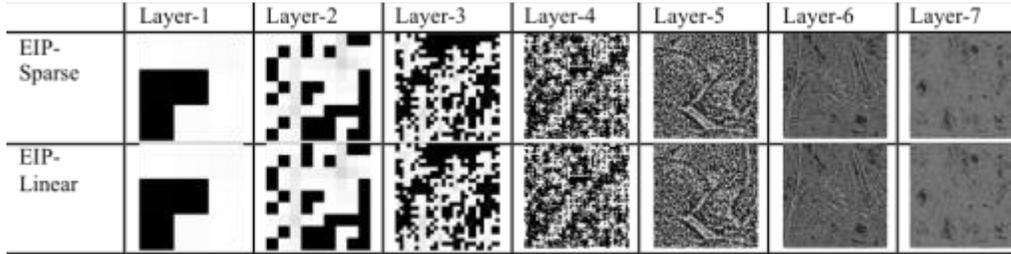


Figure 5. Laplacian feature maps

ACKNOWLEDGMENT

The research leading to these results has received funding from the European community's Seventh Framework Programme under grant agreement No. 607691 (SLANDAIL).

REFERENCES

- [1] Mitchell, Melanie. An introduction to genetic algorithms. MIT press, 1998.
- [2] Gupta, Madan, Liang Jin, and Noriyasu Homma. Static and dynamic neural networks: from fundamentals to advanced theory. John Wiley & Sons, 2004.
- [3] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554.
- [4] Marr, David, et al. "A computational theory of human stereo vision." *From the Retina to the Neocortex*. Birkhäuser Boston, 1991. 263-295.
- [5] Klette, Reinhard, and Azriel Rosenfeld. *Digital geometry: geometric methods for digital picture analysis*. Elsevier, 2004.
- [6] Sheridan, P., Hintz T. and Alexander, D., "Pseudo-invariant Image Transformations on a Hexagonal Lattice," *Image and Vision Computing*, vol. 18, pp. 907-917, 2000.
- [7] Jing, Min, Scotney, BW, Coleman, SA and McGinnity, TM (2015) A Novel Spiral Addressing Scheme for Rectangular Images. In: *Machine Vision Applications*, Tokyo. IAPR.
- [8] Róka, A., Csapó, Á., Reskó, B., & Baranyi, P. (2007). Edge detection model based on involuntary eye movements of the eye-retina system. *Acta Polytechnica Hungarica*, 4(1), 31-46.
- [9] Marr, David, and Ellen Hildreth. "Theory of edge detection." *Proceedings of the Royal Society of London B: Biological Sciences* 207.1167 (1980): 187-217.
- [10] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." *Alvey vision conference*. Vol. 15. 1988.
- [11] Coleman, SA, Scotney, BW and Gardiner, B (2011) Biologically Motivated Feature Extraction. In: *16th ICIAP, Ravenna, Italy*. Springer. Vol 6978/2 10 pp.

IV. CONCLUSION

This paper presents an efficient approach to fast image processing that utilises the eye tremor and squirrel architecture first presented in [7]. By simply modifying the image representation to represent the set of 1D images as a 2D structure and perform convolution in the vertical direction rather than horizontally, we demonstrate that fast algorithmic run-times can be achieved compared with the traditional approach of standard 2D convolution. These results are significant as future work will involve processing video data for robot vision based applications in which, in order to coarsely process the video, we need only sparsely process each frame and reconstruct a coarse feature map at each frame, built from the 9 sparsely processed previous frames. This will represent a significant speed up for video processing. For completeness the "peppers" image was used as a test case in the experiments as illustrated in Figure 5. As a note the native resolution of this image is 512×512 pixels which allows for a maximum layer 4 scaling. To allow for tests at layer 6 and 7 scales, the image was resized to 2189×2189 pixels using bilinear interpolation.