Multiscale Edge Detection using a Finite Element Framework for Hexagonal Pixel-based Images

B. Gardiner, *Member, IEEE*, SA. Coleman, and BW. Scotney

***Abstract*—In recent years the processing of hexagonal pixel-based images has been investigated, and as a result, a number of edge detection algorithms for direct application to such image structures have been developed. We build on this research by presenting a novel and efficient approach to the design of hexagonal image processing operators using linear basis and test functions within the finite element framework. Development of these scalable first order and Laplacian operators using this approach presents a framework both for obtaining large-scale neighbourhood operators in an efficient manner and for obtaining edge maps at different scales by efficient reuse of the 7-point Linear operator. We evaluate the accuracy of these proposed operators and compare the algorithmic performance using the efficient linear approach with conventional operator convolution for generating edge maps at different scale levels.**

***Index Terms*—Hexagonal image processing, edge map scaling, scalable operator, finite element.**

I. INTRODUCTION

DIGITAL image representation traditionally involves the use of a rectangular image lattice, and therefore techniques for processing such images, e.g. edge detection, have been developed for direct use on rectangular pixel-based images. An alternative concept that has been investigated is the use of hexagonal pixels for image representation, introducing the area of hexagonal image processing. Hexagonal lattices have been explored for approximately forty years [14], [34], [37], making hexagonal sampling attractive for practical applications, although only recently have attempts been made to apply processing techniques directly to hexagonal images. An overview of the advancements in edge detection techniques can be found in [10], including an approach of Canny edge detection on a hexagonal grid. Other edge detection methods on a hexagonal grid have been developed in [8], [29], [34], [35].

One of the prominent areas within image processing applications is the area of machine vision, where research is continuously being conducted to achieve improved vision systems for machine and robot control. As machine vision systems are often concerned with how fast processing can be completed, improving the computational efficiency of image processing tasks has become a dominant issue with the ultimate goal of real-time processing. Image processing tasks, in particular edge detection, are computationally expensive and to date techniques such as relaxation labelling [20], automatic scale selection [26], [27], watershed pyramids [23] and scale variant image pyramids [36] have been developed to address this issue. Such algorithms assume the use of traditional rectangular pixel-based images.

The use of hexagonally structured operators is computationally efficient when compared with square edge detection operators, due to approximately 13.5% fewer hexagonal pixels being needed to represent the same image resolution compared with a square structured grid [28]. In addition hexagonal operators typically contain fewer operator values than the corresponding square operators, thus achieving a significant overall reduction in computation. For example, for a given 256×256 image, removing boundary pixels, 63504 pixels will be processed. Using a 3×3 operator there will be 63504×25 multiplications totalling 1,587,600. If the same image is re-sampled onto a hexagonal based image there will be 55566 pixels processed by an equivalent hexagonal gradient operator containing only 19 values. Therefore there will be only 1,055,754 multiplications, corresponding to 66.5% of the computation required to generate a similar edge map using an equivalent traditional square pixel-based image.

To date, research on processing hexagonal images includes areas such as image reconstruction [24], [37], hexagonal filter banks [21], [35], blue-noise halftoning [19], image segmentation [1], [2], [18] and facial recognition [25], [32]. Many edge detection algorithms that exist for conventional images are based on components strongly aligned with the horizontal and vertical axes, and hence they are not readily adaptable to a hexagonal lattice. Only a small number of edge detection operators have been designed for use on hexagonal images, namely Prewitt [34] and Sobel [28], [39] operators, which have been modified from existing edge detection operators designed for use on conventional rectangular grids. He et al. has investigated using these operators for edge detection based on a virtual processing environment [15], [16]. In addition Davies [8] and Shima [33] have proposed derivative operators designed explicitly for use on hexagonal images. Davies' edge detection operator is comprised of two

Manuscript submitted September XX, 2015.

B. Gardiner is with the School of Computing and Intelligent Systems, University of Ulster, Londonderry, Northern Ireland, UK., Phone: +442871675081; e-mail: b.gardiner@ulster.ac.uk

S.A. Coleman is with the School of Computing and Intelligent Systems, University of Ulster, Londonderry, Northern Ireland, UK., Phone: +442871675030; e-mail: sa.coleman@ulster.ac.uk

B.W. Scotney is with School of Computing and Information Engineering, University of Ulster, Coleraine, Northern Ireland, UK., Phone: +44287 70124648; e-mail: bw.scotney@ulster.ac.uk

masks designed on the Cartesian axes, enabling the conventional formulae for computing both the gradient magnitude and direction of an edge to be used. Davies computed these masks by comparing the relationship between three directional masks and using vector addition to generate masks in the 0^o and 90^o directions. More recently Shima designed tri-directional hexagonal operators derived within the frequency domain of hexagonal images. Using Fourier transforms, hexagonal operators were generated for explicit use on hexagonal images. This hexagonal operator design is quite computationally expensive and does not offer flexibility to readily scale the operator neighbourhood to obtain larger operator sizes.

In [3], [6], [11], [13] Gardiner et al., presented an approach to edge detection operator construction that incorporated operator scalability using Gaussian test functions. In this paper we build on this work by presenting a novel and efficient approach to the design of hexagonal image processing operators using linear basis and test functions within the finite element framework. Using a linear test function within the operator design is a simple way of approximating the Gaussian function used in [11], [13]. Section 2 discusses the approach used to generate hexagonal pixel-based images. Development of the scalable first order and Laplacian operators is presented in Section 3, with their accuracy performance evaluated in Section 4. In Section 5 we present a computationally efficient approach to hexagonal based edge detection by demonstrating the need to develop only a 7-point linear operator using the finite element approach to generate a linear operator level representation for obtaining operators at larger scales or to generate an edge map scaling approach whereby each level is an edge map obtained at a different scale generated from the edge map response of a 7-point linear operator. The efficient performance of this approach is demonstrated by comparing run-times for obtaining edge maps by applying operators at various scales directly to hexagonal images with the edge map scaling approach. Finally a summary and details of future work is presented in Section 6.

II. Hexagonal Image Simulation

A factor that has limited the use of hexagonal images for image representation is a lack of hardware to capture and display images structured on a hexagonal lattice. A hexagonal image can be obtained by resampling a standard square pixel-based image using an appropriate resampling technique. In Gardiner et al. [12], a comparative evaluation was completed to determine the most appropriate resampling technique to generate hexagonal pixel-based images, evaluating those discussed in [14], [28], [37], [38]. Based on the evaluation results obtained in [12], we have chosen to use the resampling technique in [37] throughout this work. To avoid the loss of image resolution that may result from other resampling approaches, Wu et al. [37] partition each original pixel into a nxn block of sub-pixels having the same intensity as the original pixel. As illustrated in Fig. 1(a), each pixel in the original image is represented by a 7×7 block of equal intensity in the new sub-pixel image. Each hexagonal pixel is then created by clustering 56 of these sub-pixels (Fig. 1 (b)), with its intensity calculated as the average intensity of the 56 sub-pixels as shown in Fig. 1 (c).

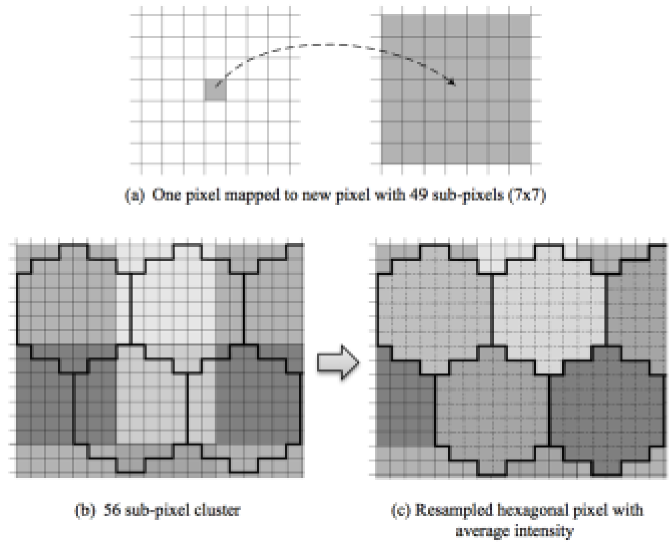


Fig. 1. Hexagonal pixel at sub-pixel level

III. Operators for Processing Hexagonal Images

In order to develop scalable and efficient gradient operators for use on hexagonally structured images, we use the flexibility offered by the finite element framework. The use of the finite element framework for the derivation of image processing techniques has been successfully demonstrated in work such as [31] where the finite element framework was implemented on a traditional rectangular pixel array to develop and analyse near-circular edge detectors, [22] where the framework was used to develop a scale invariant interest point detector, and in [5] where the framework is adapted for use on range and intensity images. When used with a hexagonal pixel array the six-fold symmetry of the naturally occurring computational grid of equilateral triangular elements enables particularly efficient implementation through use of rotational symmetries. This means that, unlike other hexagonal methods, not only are we able to provide a systematic technique for scaling operators on a hexagonal grid, we can do this with low computational complexity – even more so than on a rectangular grid due to the increased degree of rotational symmetry present in the computational mesh.

A. Hexagonal Image Representation

In order to apply a finite element based approach to image processing tasks, the hexagonal image must be represented as a discrete function. Typically an image can be represented by an array of samples of a continuous function u of image intensity on a domain Ω. Nodes are placed in the centre of each hexagonal pixel within the hexagonal image domain. These nodes are the reference points for finite element computation throughout the domain Ω. Interconnecting each of these nodes in the image domain produces the edges of the triangular structured finite elements. A representation of the finite element mesh of equilateral triangular elements is shown in Fig. 2. The broken lines represent the hexagonal pixels throughout the image array. A finite element mesh now exists where a nodal numbering scheme 1,…,N is employed globally throughout the entire image.

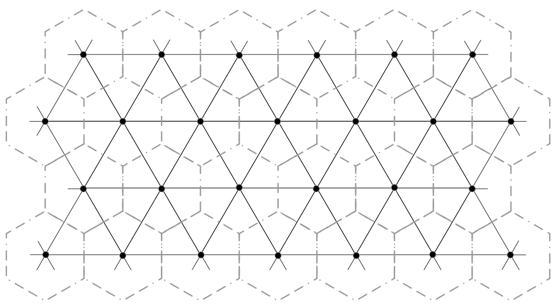


Fig. 2. Finite element mesh of equilateral triangular elements

A common approach to addressing pixels on a hexagonal grid is to use a three axes co-ordinate system that utilises the three axes of symmetry of a hexagon. An advantage is that moving from the centre of one pixel to the centre of a neighbouring pixel requires a unit shift along only one axis (see Fig. 3). A disadvantage is that any pixel centre does not have a unique address.

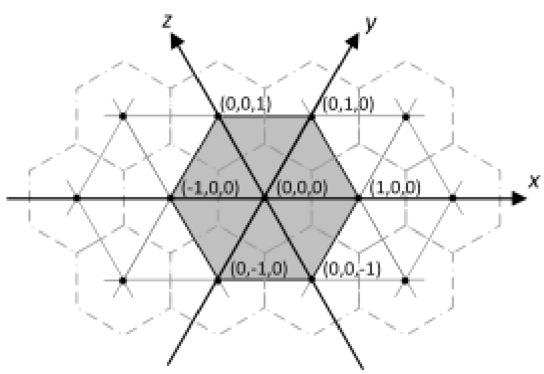


Fig. 3. Three axes hexagonal co-ordinate system

An alternative approach is to select two of the three skewed axes providing unique representation for any point in the image plane. We choose the x and y axis in Fig. 3.

Using the two axes co-ordinate system, the continuous image intensity function u is approximated on the domain Ω by the function U from a finite dimensional function space $S^h(\Omega)$. A set of functions $\varphi_i(x,y), i = 1,\dots,N$, is chosen as a basis for S^h. Such a basis can be formed by associating any node i, with co-ordinates (x_i, y_i), with a piecewise linear basis function which has the properties

$$\varphi_i(x_j, y_j) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \quad (1)$$

$\varphi_i(x,y)$, is thus a "tent-shaped" function with support restricted to a small neighbourhood centred on node i consisting of only those elements that have node i as a vertex. A 3D representation of the basis function is shown in Fig. 4.

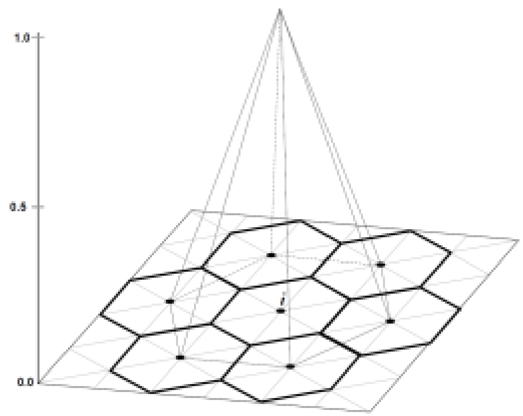


Fig. 4. 3D representation of basis function

We can represent any function $F(x,y) \in S^h$ by a set of co-efficients $\{F_1,\dots,F_N\}$ in the form

$$F(x,y) = \sum_{j=1}^{N} F_j \varphi_j(x,y) \quad (2)$$

In particular, we can use this form to approximately represent the image u by a function

$$U(x,y) = \sum_{j=1}^{N} U_j \varphi_j(x,y) \quad (3)$$

in which the parameters $\{U_j\}$ are mapped from the hexagonal image intensity values. The approximate image representation is therefore a simple piecewise linear function on each triangular element in the finite element mesh.

B. *Hexagonal Operator Design*

To develop an operator that is implemented on a specific neighbourhood, a test function is selected and used within the weak form of the operator. This involves numerical integration of the test function with the image derivative over the neighbourhood; operators at different scale can be achieved by selecting differently sized neighbourhoods and correspondingly scaled test functions. Each test function is restricted to have support over the neighbourhood, centred on node i. In general the size of the neighbourhood Ω_i^σ may be related explicitly to the scale parameter σ [9], as illustrated by the first three sizes of neighbourhoods shown in Fig. 5, i.e., 7-point (H_3), 19-point (H_5), and 37-point (H_7) hexagonal neighbourhood operators, which are approximately equivalent in size to the 3x3, 5x5 and 7x7 conventional rectangular operators, respectively. The parameter σ corresponds to the "operator width": for $H_n, \sigma = \sigma = \frac{n-1}{2}$.

It is evident from Fig. 5 that a 7-point hexagonal operator is computed on a six element neighbourhood, a 19-point operator is computed over 24 elements, and a 37-point operator has 54 elements present in its neighbourhood. Development of scalable first order derivative and Laplacian operators is presented in Section 3.2.1 and Section 3.2.2 respectively.

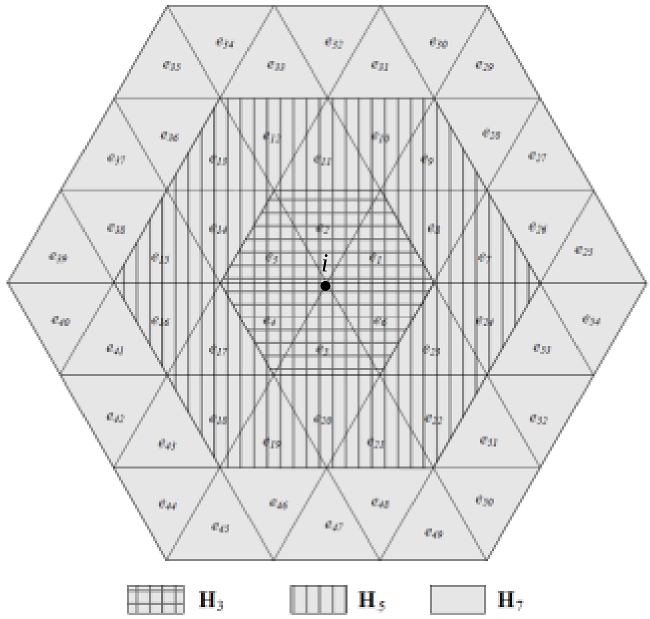


Fig. 5. Hexagonal operator neighbourhoods for operator sizes H3, H5 and H7

1) First Order Derivative Operator

Computation of the first order derivative operator follows a standard finite element approach that uses a weak form of the derivative. The weak form requires the image function to be once differentiable in the sense of belonging to the Hilbert space $H^1(\Omega)$ over the image domain Ω. That is, $u = u(x, y)$ is such that the integral $\int_\Omega (|\underline{\nabla} u|^2 + u^2)\, d\omega$ is finite, where ω is the Lebesgue measure on Ω, and $\underline{\nabla} u$ is the image gradient. Thus by requiring that $u \in H^1$, the problem is to find the weak form of the directional derivative of the image on the image domain Ω, namely

$$E(U) = \int_\Omega \underline{\mathrm{b}} \cdot \underline{\nabla} u v \, d\omega \tag{4}$$

where $v \in H^1$, and $\underline{b}$ is the unit direction vector. Altering the direction of the unit vector in the weak form would produce an operator design for use on alternative co-ordinate systems e.g. using vectors in the hexagonal x and y axes directions produces weak form for use on a hexagonal co-ordinate system.

In order to use the finite element approach to construct scalable first order operators, the weak form of the x-directional derivative is used, which is given by the functional

$$E_i^\sigma(U) = \int_{\Omega_i^\sigma} \frac{\delta u}{\delta x} \psi_i^\sigma \, d\omega_i \tag{5}$$

In developing a Linear-Linear derivative operator (i.e. both basis and test functions are linear) at the smallest scale we use the Galerkin formulation, i.e. the test functions ψ_i^σ used in the weak form are from the same space as those used in the image representation, i.e. $\psi_i^\sigma = \varphi_i$.

To illustrate the implementation of a first order Linear-Linear operator (L), we build a 7-point hexagonal operator as shown in Fig. 6. At the smallest scale, the neighbourhood Ω_i^σ ($\sigma = 1$) covers a set of six elements $\{e_m\}$ where the piecewise linear basis function φ_i is associated with the central node i which shares common support with the surrounding six basis functions φ_j. On each element e_m a local co-ordinate reference system for a general equilateral triangular element is used with one of the nodes α, *b*, λ, corresponding to the central node i, as illustrated in Fig. 6 for element e_1 of a neighbourhood.

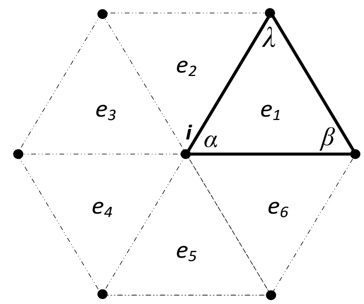


Fig. 6. Elements within neighbourhood $\Omega_i^\sigma, \sigma = 1$

To create a derivative operator over a neighbourhood Ω_i^σ, we substitute the image representation in (3) into the functional $E_i^\sigma(U)$, which yields

$$E_i^\sigma(U) = \sum_{j=1}^{N} K_{ij}^\sigma U_j \tag{6}$$

where K_{ij}^σ are the entries in the $N \times M$ global matrix K^σ given by

$$K_{ij}^\sigma = \sum_{m | e_m \in S_i^\sigma} k_{ij}^{m,\sigma} \tag{7}$$

And $k_{ij}^{m,\sigma}$ is the element integral

$$k_{ij}^{m,\sigma} = \int_{\Omega_i^\sigma} \frac{\delta \varphi_j}{\delta x} \varphi_i \quad d\omega_i^\sigma \tag{8}$$

The integral shown in equation (8) is computed only over the neighbourhood Ω_i^σ rather than the entire image domain Ω since φ_i has support restricted to Ω_i^σ. For each of the six triangular elements within the neighbourhood, a triangular element operator is generated whose entries then map directly to the corresponding locations within the 7-point operator neighbourhood. For example, consider element e_1 shown in Fig. 6. On this element the basis functions φ_j for $j = \alpha, \beta, \lambda$ share common support with φ_i. Hence the first derivative triangular element operator is computed as

$$k_i^{1,\sigma} = \begin{bmatrix} & k_{i\lambda}^{1,\sigma} & \\ k_{i\alpha}^{1,\sigma} & & k_{i\beta}^{1,\sigma} \end{bmatrix} \tag{9}$$

where $k_{ij}^{1,\sigma}$ is computed using the element integral in (8) with the linear basis functions

$$\varphi_\alpha = 1 - x - y \qquad \varphi_\beta = x \qquad \text{and} \qquad \varphi_\lambda = y \tag{10}$$

which when differentiated with respect to x give

$$\frac{\delta\varphi_\alpha}{\delta x} = -1 \qquad \frac{\delta\varphi_\beta}{\delta x} = 1 \qquad \text{and} \qquad \frac{\delta\varphi_\lambda}{\delta x} = 0 \tag{11}$$

Using the hexagonal co-ordinate system presented in Fig. 3, $k_{ij}^{1,\sigma}$ is represented as

$$k_{ij}^{1,\sigma} = \int_0^1 \int_0^{1-x} \frac{\delta\varphi_j}{\delta x} \varphi_i |J| \, dy \, dx \tag{12}$$

where the Jacobian J has the value $\sqrt{3}/2$.

To demonstrate the linear element computation, consider nodes α, β, λ in e_1 . $k_{i\alpha}^{1,\sigma}$ may be written as

$$k_{i\alpha}^{1,\sigma} = \int_0^1 \int_0^{1-x} (-1)(1-x-y)\frac{\sqrt{3}}{2} dy \, dx \tag{13}$$

$$= \int_0^1 \int_0^{1-x} (-1+x+y)\frac{\sqrt{3}}{2} dy \, dx \tag{14}$$

Similarly $k_{i\beta}^{1,\sigma}$ may be written as

$$k_{i\beta}^{1,\sigma} = \int_0^1 \int_0^{1-x} (1)(1-x-y)\frac{\sqrt{3}}{2} dy \, dx \tag{15}$$

$$= \int_0^1 \int_0^{1-x} (1-x-y)\frac{\sqrt{3}}{2} dy \, dx \tag{16}$$

and $k_{i\lambda}^{1,\sigma}$ may be written as

$$k_{i\lambda}^{1,\sigma} = \int_0^1 \int_0^{1-x} (0)(1-x-y)\frac{\sqrt{3}}{2} dy \, dx = 0 \tag{17}$$

The element operators for the six elements may thus be computed as:

$$k_i^{1,\sigma} = \begin{bmatrix} & 0 & \\ -a & & a \end{bmatrix}, \quad k_i^{2,\sigma} = \begin{bmatrix} -a & & a \\ & 0 & \end{bmatrix},$$
$$k_i^{3,\sigma} = \begin{bmatrix} & 0 & \\ -a & & a \end{bmatrix}, \quad k_i^{4,\sigma} = \begin{bmatrix} -a & & a \\ & 0 & \end{bmatrix}, \tag{18}$$
$$k_i^{5,\sigma} = \begin{bmatrix} & 0 & \\ -a & & a \end{bmatrix}, \quad k_i^{6,\sigma} = \begin{bmatrix} -a & & a \\ & 0 & \end{bmatrix}.$$

where a = 0.1443. These element operators can then be appropriately assembled to generate a 7-point Linear-Linear hexagonal operator. This is achieved by carrying out a standard finite element assembly procedure, providing the neighbourhood structure for the x-derivative operator shown in Fig. 7.

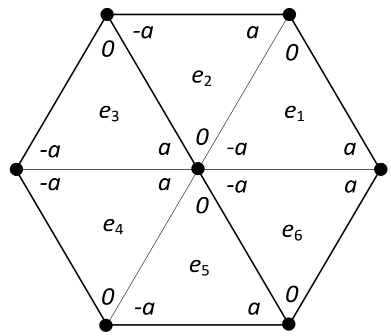


Fig. 7. Element operators within neighbourhood W_i^s

Combination of the element operators in Fig. 7 yields

$$L_3^x = \begin{bmatrix} & (0-a) & & (0+a) & \\ (-a-a) & & (-a+a-a+a) & & (a+a) \\ & (0-a) & & (0+a) & \end{bmatrix} \tag{19}$$

$$= \begin{bmatrix} & -a & & a & \\ -2a & & 0 & & 2a \\ & -a & & a & \end{bmatrix} \tag{20}$$

Substituting the value of a into equation (20) completes the x-derivative operator as

$$L_3^x = \begin{bmatrix} & -0.144 & & 0.144 & \\ -0.288 & & 0 & & 0.288 \\ & -0.144 & & 0.144 & \end{bmatrix} \tag{21}$$

By rotating the x-directional operator anti-clockwise by 60^o and 120^o, the y- and z- directional operators can be readily obtained, respectively, as

$$L_3^y = \begin{bmatrix} & a & & 2a & \\ -a & & 0 & & a \\ & -2a & & -a & \end{bmatrix} \quad L_3^z = \begin{bmatrix} & 2a & & a & \\ a & & 0 & & -a \\ & -a & & -2a & \end{bmatrix} \tag{22}$$

When calculating the gradient response for tri-directional derivative operators, redundancy is introduced due to the relationships by rotation between the three operators, and the gradient magnitude can be represented using only operators L_n^x and L_n^z as

$$|G_n| = \frac{2}{\sqrt{3}}\sqrt{\left(L_n^x\right)^2 + \left(L_n^z\right)^2 + L_n^x . L_n^z} \tag{23}$$

where n represents the size of neighbourhood of the operator.

2) Laplacian Operator

In developing Laplacian Linear-Linear hexagonal operators (LL) we can combine the approximate image representation and test function to generate an approximate representation of the weak form of the x-component of the Laplacian operator, which is represented by the functional

$$R_i^\sigma(U) = \int_\Omega \frac{\delta U}{\delta x}\frac{\delta \varphi_i}{\delta x}\, d\omega \tag{24}$$

Substitution of the hexagonal image representation

$$U(x,y) = \sum_{j=1}^{N} U_j \varphi_j(x,y) \tag{25}$$

into equation (24) gives

$$R_i^\sigma = \sum_{j=1}^{N} K_{ij}^\sigma U_j \tag{26}$$

where K_{ij}^σ are the entries in the global matrix K^σ given by

$$K_{ij}^\sigma = \int_{\Omega_i^\sigma} \frac{\delta \varphi_j}{\delta x}\frac{\delta \varphi_i}{\delta x}\, d\omega_i^\sigma \tag{27}$$

Again, the integral shown in equation (27) is computed only over the neighbourhood Ω_i^σ as opposed to the entire image domain Ω as the linear test function φ_i has support restricted to Ω_i^σ. This 7-point operator (LL_3^x) has the structure shown in equation (15), with the values of the co-efficients a and b being 0.866 and 1.732, respectively.

$$\mathrm{LL}_3^x = \begin{bmatrix} & 0 & & 0 & \\ -a & & b & & -a \\ & 0 & & 0 & \end{bmatrix} \tag{28}$$

The Laplacian hexagonal operator LL_3 can be expressed as the sum of the Laplacian components x, y and z denoted as LL_3^x, LL_3^y and LL_3^z respectively. In order to obtain LL_3^y and LL_3^z, the co-efficients of LL_3^x must be rotated anti-clockwise by 60^o and 120^o respectively. The Laplacian operator is then obtained by appropriate summation of these operators $\mathrm{LL}_n = \frac{2}{3}(\mathrm{LL}_3^x + \mathrm{LL}_3^y + \mathrm{LL}_3^z)$, where n represents the size of neighbourhood of the operator. In the case of n=3, we obtain

$$\mathrm{LL}_3 = \begin{bmatrix} & -0.577 & & -0.577 & \\ -0.577 & & 3.462 & & -0.577 \\ & -0.577 & & -0.577 & \end{bmatrix} \tag{29}$$

IV. Performance Evaluation

To evaluate the Linear-Linear operators developed in this work we initially compare the performance of the 7-point Linear-Linear operator with other existing hexagonal operators such as the well-known Prewitt, Sobel, Davies operators and the recent technique of Shima. In making this selection we note that the Sobel operator corresponds to the simplest form of the Canny operator without any additional post-processing steps. As the Linear-Linear operators are using a linear approximation of the Gaussian smoothing incorporated in the operators presented in [11], we further evaluate the operators by comparing the accuracy performance of these Gaussian based operators with the performance of the proposed operators. We have chosen two evaluation techniques: a quantitative method, the Figure of Merit evaluation [30], and a qualitative method, the Robust Visual Method [17]. We have adapted the well-known Figure of Merit (FoM) algorithm to enable evaluation using synthetic hexagonal pixel-based images of curved edges as well as straight edges at various orientations. The Robust Visual Method is used to visually evaluate operator edge maps, based on human evaluators rating the visual integrity of edge maps generated by different operators.

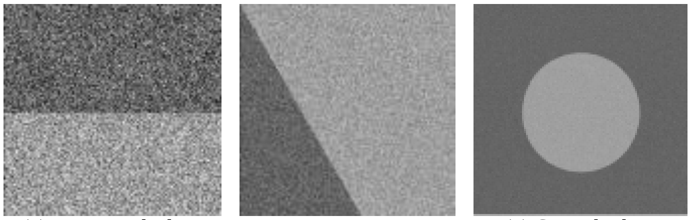
A. *Figure of Merit Evaluation*

We initially evaluated the output responses of these operators using the Figure of Merit evaluation technique. This technique considers three major areas of error associated with the determination of an edge: missing valid edge points; failure to localise edge points; classification of noise fluctuations as edge points. In addition to these considerations, when measuring edge detection performance, edge detectors that produce smeared edge locations should be penalised, whilst those that produce edge locations that are localised should be awarded credit. Hence Pratt introduced the Figure of Merit technique as one that balances the three types of error above, defined as

$$R = \frac{1}{\max(I_A, I_I)}\sum_{i=1}^{I_A}\left(\frac{1}{1+\alpha d^2}\right) \tag{30}$$

where I_A is the actual number of edge pixels detected, I_I is the ideal number of edge pixels, d is the separation distance of a detected edge point normal to a line of ideal edge points, and α is a scaling factor. The Figure of Merit is normalised such that R takes values between 0 and 1, where 1 represents a perfectly detected edge. The scaling factor, α, is most commonly chosen to be 1/9, although this value may be adjusted to penalise edges that are localised but offset from the true edge position. Since knowledge of the actual edge location is necessary, this method can only be used on synthetic images.

To provide a realistic environment to compare operator responses, the Figure of Merit (FoM) technique is used on images with varying signal-to-noise ratios (SNR), where $SNR = h^2/\sigma_n^2$, h is the height of the step edge and σ_n^2 is the variance of the noise. Synthetic images for Figure of Merit measurements typically contain horizontal, vertical or oriented edges. However, one proposed advantage of hexagonal pixel-based images is their ability to accurately represent curves in real images. Therefore, we extend the standard use of the Figure of Merit technique to incorporate the measure of detected curved edges. The synthetic test images used for evaluation are generated using h=58 with SNR = 100, 50, 20, 10, 5 and 1 and contain a horizontal edge, an edge oriented at 60^0 or a curved edge (examples of which are presented in Fig. 8) Five sets of test images were generated for each edge type, at each SNR (totalling 90 test images). The FoM was calculated for each operator over the test image set and averaged to obtain an accurate Figure of Merit result.



(a) Horizontal edge (SNR=1) (b) 60^0 oriented edge (SNR=10) (c) Curved Edge (SNR=100)

Fig. 8. Example images for use in Figure of Merit

Fig. 9 to Fig. 11 inclusive show Figure of Merit results comparing the 7-point Linear-Linear operator, denoted by L3, with existing hexagonal operators of the same neighbourhood size, i.e. Sobel, Davies, Prewitt and Shima operators. The results illustrate that the proposed L3 operator has increased accuracy over the Prewitt operator in all evaluated edge directions, while demonstrating that the L3 operator achieves the same performance accuracy as the Sobel, Davies and Shima operators. This is due to the Sobel, Davies and Shima operators being equivalent to the L3 operator in relation to their weight proportions, i.e., the weight values of the operator are proportioned to achieve smoothing by giving greater importance to the centre weight values. However, the L3 operator design facilitates the implementation of larger sizes of operators using the flexibility of the finite element framework for neighbourhood operator scaling, which is discussed in Section 3.

As the Linear-Linear operators use a linear approximation of the Gaussian smoothing incorporated in the operators presented in [11], we further evaluate the operators by comparing the accuracy performance of these Gaussian based operators with the performance of the proposed operators. For comparison we consider the three smallest operator scales, i.e. 7-point, 19-point and 37-point neighbourhood operators. Fig. 12 to Fig. 14 inclusive show Figure of Merit results comparing the set of scaled Linear-Linear operators, denoted by L3, L5 and L7 with previously developed Linear-Gaussian hexagonal operators, denoted by LG3, LG5 and LG7.

Results show that, in most cases, our first order Linear-Linear operators perform as well as the Linear-Gaussian operators. The 7-point Linear-Linear operator (L3) generates equivalent results to the 7-point Linear-Gaussian (LG3), whereas the 19-point and 37-point Linear-Gaussian operators perform slightly better compared with the equivalent sized Linear-Linear operators on images with high levels of noise. This slight decrease in performance would be expected of the family of Linear-Linear operators as the linear function used when constructing the operators is only an approximation to the Gaussian function. However, this minor difference in output performance is counteracted by the Linear-Linear operator structure enabling methods to efficiently obtain edge detection results by either linear operator scaling or linear edge map scaling, as discussed in Section 5. It also should be noted that work previously published by the authors [7] has compared the performance of the Linear-Gaussian operators with conventional square operators at multiple scales, demons-

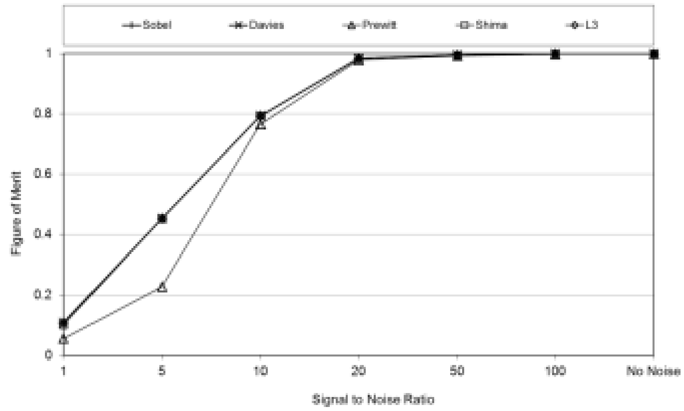


Fig. 9. Comparing with existing hexagonal operators using a 60^o oriented edge

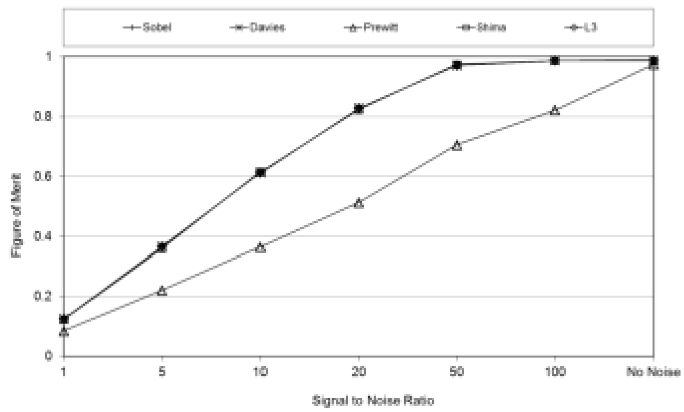


Fig. 10. Comparing with existing hexagonal using a Curved edge

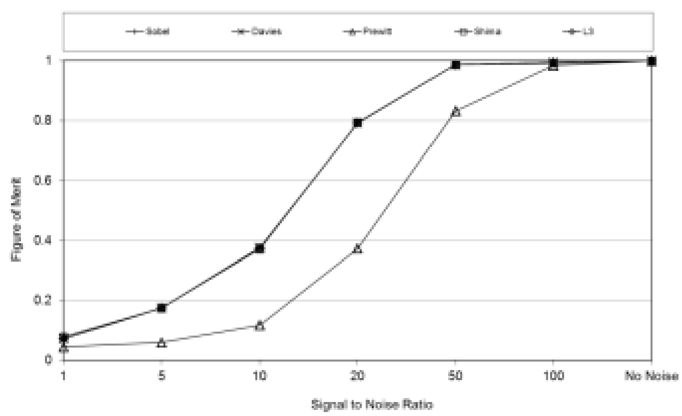


Fig. 11. Comparing with existing hexagonal using a vertical edge

-trating that the performance of the Linear-Gaussian operators were comparable to, and in some cases slightly superior than, the equivalent use of typical operators on standard square pixel-based images. Therefore comparing the performance of the proposed Linear-Linear operators with the Linear-Gaussian operators also highlight how the proposed operators produce comparable results to those obtained from conventional square operators.

Consider also the Laplacian Linear-Linear operators. The use of small Laplacian operators, e.g. 3x3 or equivalent opera-

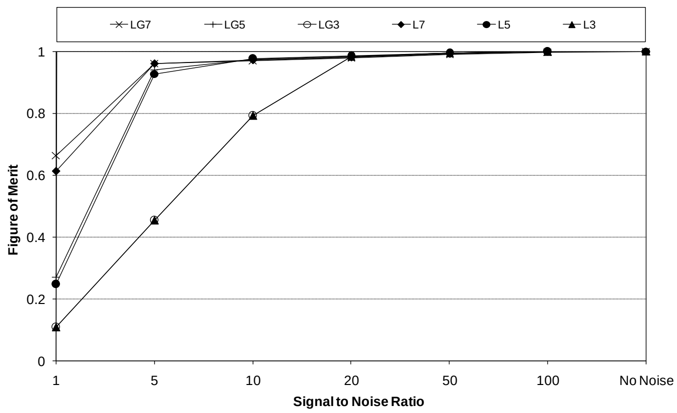


Fig. 12. Comparing with previously developed Linear-Gaussian hexagonal operators using a 60^{o} oriented edge

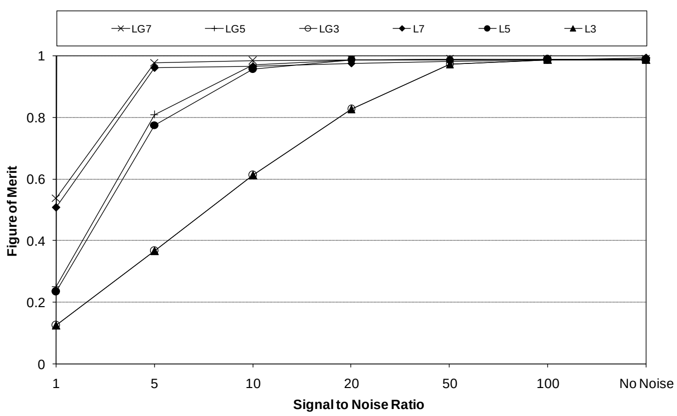


Fig. 13. Comparing with previously developed Linear-Gaussian hexagonal operators using a Curved edge

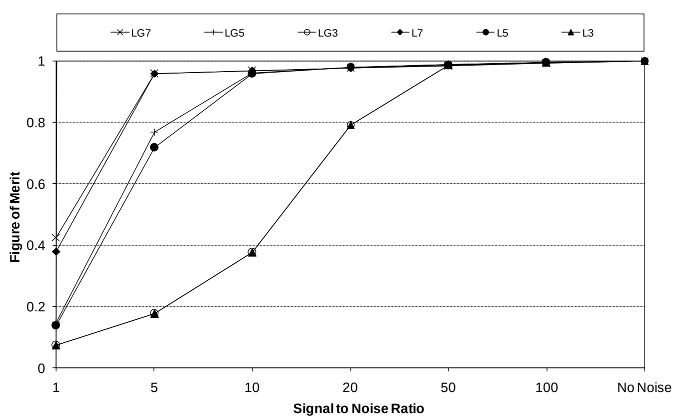


Fig. 14. Comparing with previously developed Linear-Gaussian hexagonal operators using a vertical edge

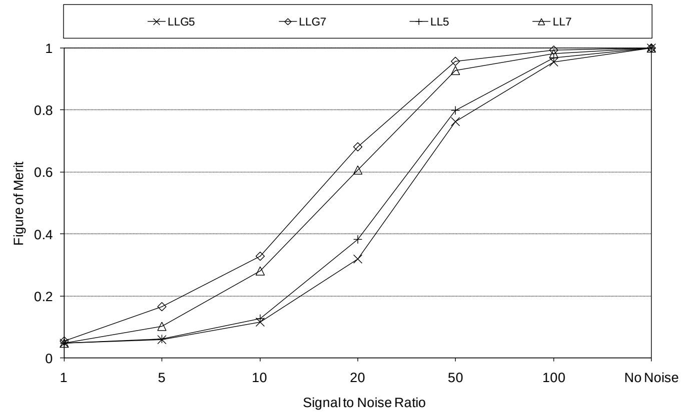


Fig. 15. Comparing with previously developed Laplacian Linear-Gaussian operators using a 60^{o} oriented edge

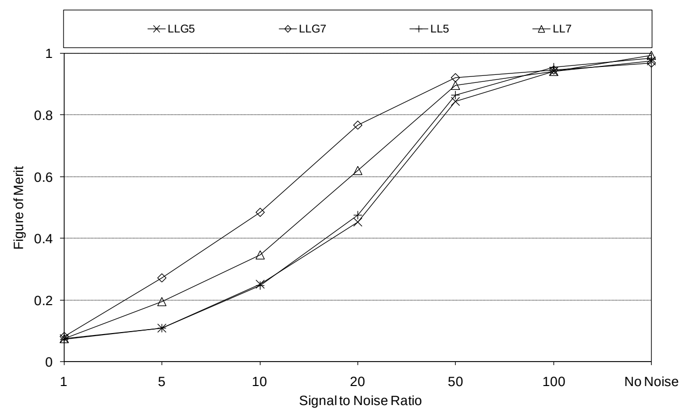


Fig. 16. Comparing with previously developed Laplacian Linear-Gaussian operators using a Curved edge

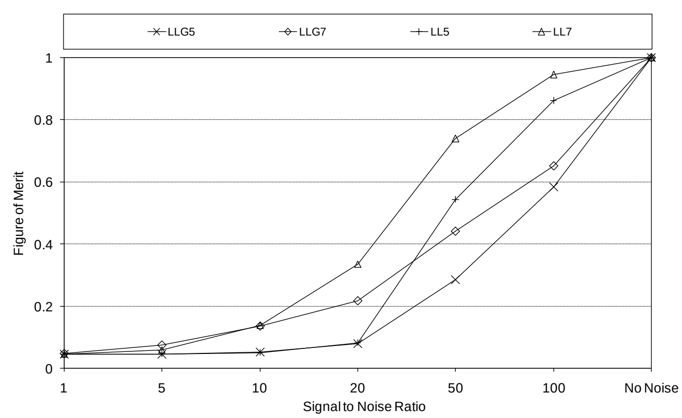


Fig. 17. Comparing with previously developed Laplacian Linear-Gaussian operators using a Vertical edge

-tors, is not common due to the general performance of such operators being poor in the presence of noise. Therefore, evaluation is presented in this section using two operator sizes,19- and 37-point, in order to comparatively evaluate the Laplacian Linear-Linear operators with existing Laplacian hexagonal operators [4]. Fig. 15 to Fig. 17 inclusive show Figure of Merit results comparing the Linear-Linear and Linear-Gaussian hexagonal operators of equivalent sizes using the three oriented edge types used above.

When comparing 19-point hexagonal operators, LL5 and LLG5, the results are quite similar for most edge orientations. The results obtained for the proposed 37-point Laplacian Linear-Linear operator (LL7) have slightly decreased accuracy when compared with the Laplacian Linear-Gaussian operator (LLG7) in the 60^{o} and curved edge orientations.

B. *Robust Visual Method Evaluation*

The robust visual method is used to evaluate operator edge maps based on human evaluators rating the visual integrity of edge maps generated by each operator.

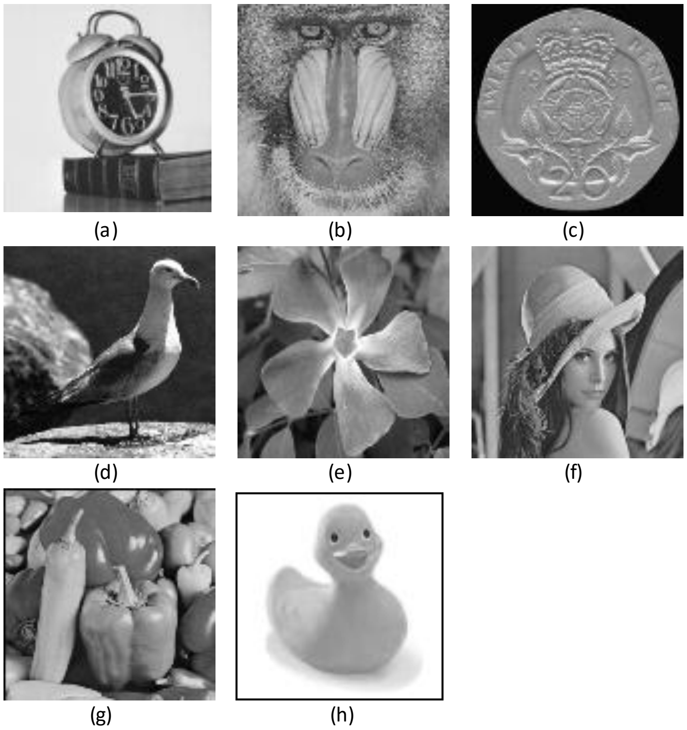


Fig. 18. Image set for robust visual method of evaluation

Most methods of evaluating operator output responses rely on the use of ground truth, but creating ground truth for real images can be time consuming and inaccurate. An advantage of the robust visual evaluation method is that it uses real images that rely on the subjective evaluation of edge maps by the human visual system and therefore does not require the use of ground truth. The real images used are selected such that they have a centrally placed object in the image foreground (Fig. 18).

In the robust visual method, the subjects rank the edge image on a scale of 1 to 7 according to how well they can recognise the centrally placed object, where 7 indicates easy recognition and 1 indicates no coherent information. The Intraclass Correlation Coefficient, calculated by the statistical measure $ICC(3,k) = \frac{BMS-EMS}{BMS}$ was used to ensure image rating consistency within the set of human subjects, where *BMS* is the mean square value of the rating, *EMS* is the total mean square error and k is the number of evaluators. In phase 1 of the technique, for any one image the human subjects rate six edge images generated by an operator over a range of thresholds. This results in the visually best edge map corresponding to each image for each operator being selected by the evaluators. In phase 2, the human subjects then rated the selected edge maps for each image on a scale of 1 to 7 in order to compare the overall performance of different operators. Again consistency was checked using $ICC(3,k)$. Initially edge maps were generated for each of the eight images at a range of thresholds using the proposed 7-point, 19-point and 37-point Linear-Linear hexagonal operators, and for comparison, equivalently sized Linear-Gaussian hexagonal operators have been applied to the same set of images. Fig. 19 shows an example edge map set (for six different threshold (T)

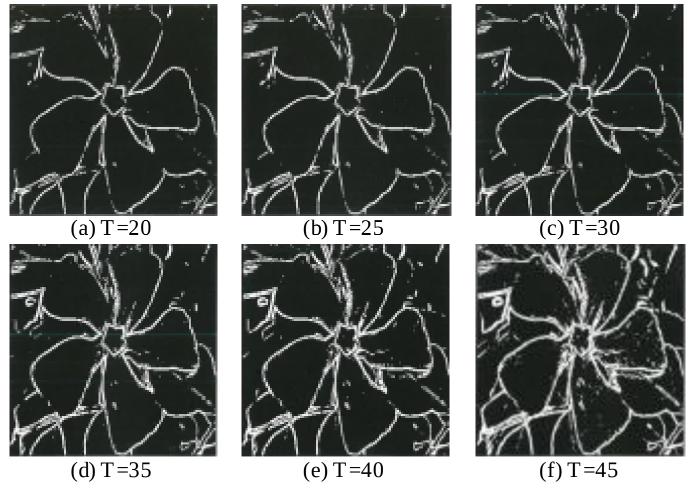


Fig. 19. An example image set for the L3 operator at various thresholds

values) for the L3 operator applied to the image shown in Fig. 18(e).

The information collected from each evaluator was analysed for consistency using an Intraclass Correlation Coefficient. The new image set was created using only the visually best edge map for each operator determined by the results obtained from the human evaluators. This image set was used to determine which operator performed best overall with respect to detecting edges. Again seven evaluators ranked the image set and consistency was tested using the Intraclass Correlation Coefficient. The mean ratings throughout the image set for each of the evaluated operators are presented in Table I. These ratings identify which operator, based on human evaluation, provides the best results when used to detect edges for a range of input images.

The results obtained indicate that the family of Linear-Linear hexagonal operators are ranked by the evaluators to perform marginally less well than the corresponding set of Linear-Gaussian operators, with the largest mean rating difference of 0.34 occurring between the 7-point operators with as little as a 0.09 mean rating difference occurring between the 37-point operators. Based on a mean rating scale between 1 and 7, these results demonstrate that visually an insignificant difference exists between the performance of the Linear-Linear operator and the Linear-Gaussian operator set.

TABLE I
MEAN RATING FOR EACH FIRST ORDER OPERATOR

FIRST ORDER OPERATOR	MEAN
7-point Linear-Linear (L3)	5.16
7-point Linear-Gaussian (LG3)	5.50
19-point Linear-Linear (L5)	5.18
19-point Linear-Gaussian (LG5)	5.46
37-point Linear-Linear (L7)	5.14
37-point Linear-Gaussian (LG7)	5.23

It is necessary to also evaluate the performance of Laplacian Linear-Linear operators and compare with previously developed Laplacian Linear-Gaussian operators [4]. The same set of real images was used as shown in Fig. 18. Again seven evaluators ranked the image set and consistency was tested using the Intraclass Correlation Coefficient. The mean ratings throughout the image set for each of the evaluated operators are presented in Table II.

The results obtained indicate that the family of Laplacian Linear-Linear hexagonal operators are ranked by the evaluators to achieve comparable results with the corresponding set of Linear-Gaussian operators, i.e. improved mean rating for the 19-point Linear-Linear operator and a slight decrease in the mean rating value for the 37-point Linear-Linear operator.

TABLE II
MEAN RATING FOR EACH LAPLACIAN OPERATOR

LAPLACIAN OPERATOR	MEAN
19-point Laplacian Linear-Linear (LL5)	5.18
19-point Laplacian Linear-Gaussian (LLG5)	4.82
37-point Laplacian Linear-Linear (LL7)	5.71
37-point Laplacian Linear-Gaussian (LLG7)	5.95

The comparable accuracy achieved throughout the evaluation methods presented for first order and Laplacian Linear-Linear operators, combined with the efficient approach in obtaining these results (discussed now in Section 5) highlights the benefits of this proposed family of scalable derivative operators for edge detection tasks.

V. EFFICIENT APPROACH TO HEXAGONAL EDGE DETECTION

We utilise the linear characteristics of the Linear-Linear operators to introduce two separate approaches to conducting edge detection efficiently on hexagonal pixel-based images. Firstly we show how it is necessary to develop only a 7-point Linear operator and then the larger scale derivative operators can be efficiently obtained via linear combinations of the 7-point operators. Secondly, efficient implementation is achieved by combination of values from the edge map at the smallest scale, and we illustrate this approach for edge detection.

It is necessary to compute the operators at only the smallest scale, as these can then be combined linearly to generate the operators at larger scales. This is because the linear test function ψ_i^σ used in the $L_n^x (n = 5, 7, ..., m)$ operator design at scale $\sigma > 1$ can be expressed as a linear combination of the test functions ϕ used to compute the L_3^x operators at the lowest scale $\sigma = 1$. To demonstrate the building of the operators, consider Fig. 20, in which we have used a radial coordinate system. Here p indicates the level of the neighbourhood nodes, i.e., $p = 0$ at the centre node, $p = 1$ for each of the surrounding nodes at the next level, etc., and q measures the angular location within a given level p. The smallest operator size (7-point operator) corresponds to neighbourhood level $p = 1$, the next operator size (19-point operator), corresponds to neighbourhood level $p = 2$ etc.

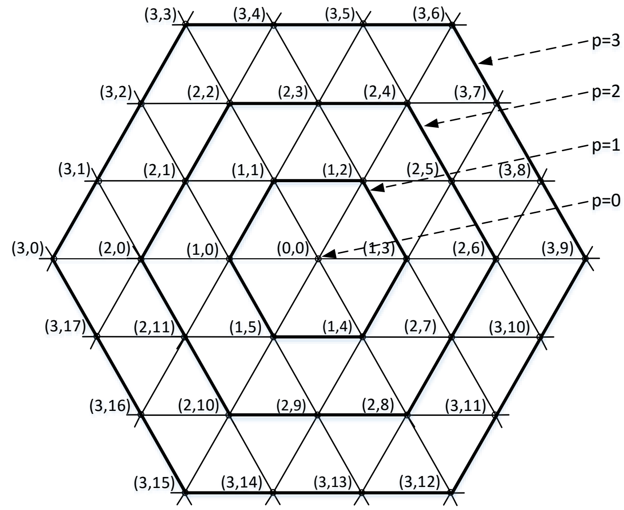


Fig. 20. Finite element mesh corresponding to 4 neighbourhood levels, p = 0, 1, 2, 3

In order to generate an L_5^x hexagonal operator, we place an L_3^x mask at the centre node of the mesh at level 0, node $(0,0)$, and $\frac{1}{2} \times (L_3^x)$ mask at the other six internal nodes at level 1. The value of $\frac{1}{2} \times (L_3^x)$ is used at each node at level 1 as the value of the linear test function ψ_i^σ for the L_5^x operator design, i.e. $\sigma = 2$, can be expressed in terms of the linear test function φ in the L_3^x operator as

$$\psi_i^2 = \phi_{(0,0)} + \frac{1}{2}\sum_{x=0}^{x=6} \phi_{(1,q)} \tag{31}$$

where $\varphi_{(p,q)}$ are the linear test function values of the L_3^x operator at nodes (p,q). These linear combinations are then used in the typical finite element assembly manner. This is illustrated in Fig. 21 for an L_5^x x-directional mask, showing L_3^x applied to the centre node and $\frac{1}{2} \times (L_3^x)$ applied to one of the nodes in level $p = 1$. Once assembly is completed, the computed nodal values correspond to the operator values at each of the points in an L_5^x x-directional mask.

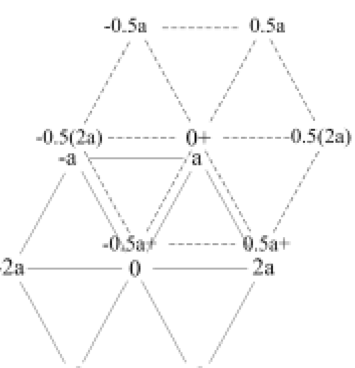


Fig. 21. Illustrating the combining of the L_3^x masks to obtain the L_5^x mask

We can generalise this procedure for any operator size L_s^x, where $s > 3$ (the initial operator). Again using the nodal system illustrated in Fig. 20, we let $K_{(p,q)}$ be the values of the

L_3^x hexagonal mask placed at each node (p,q). Consider the hexagonal operator size s (5, 7 etc.), then the radius of the approximately circular hexagonal operator O_R can be determined as

$$O_R = \frac{s-1}{2} \quad (32)$$

For each operator size s (>3), the x-directional operator, L_s^x, can be computed using the following formula:

$$L_s^x = K_{(0,0)} + \sum_{p=1}^{O_R-1} \sum_{q=0}^{6q-1} \left(\frac{O_R - p}{O_R}\right) K_{(p,q)} \quad (33)$$

where the number of levels, p, to be included is $O_R - 1$. As previously discussed, the y- and z- directional derivative operators can then be efficiently obtained by rotating the co-efficients of L_s^x anti-clockwise by 60° and 120° to obtain L_s^y and L_s^z respectively.

The characteristics of Linear operators not only permit construction of derivative operators at many scales by linear combinations of smaller operators but also provide an alternative method for obtaining the scaled edge map outputs by directly using linear combinations of edge map outputs obtained at the lowest scale.

Again this approach involves the construction of only the 7-point hexagonal operator as described in Section 3. The 7-point operator is applied to the hexagonal image to obtain an edge map. Instead of constructing a larger scale operator and convolving it with the image, we can use linear combinations of the gradient responses generated by the 7-point operator to construct the edge maps at larger scales (see Fig. 20). In this way, each edge map (M_s) is equivalent to the edge map that would be generated by our proposed linear operator at scale s. It is important to note that the image resolution is not altered at each level, but it is the operator scale that changes. The procedure of efficiently generating these edge maps is as follows. Firstly let M_1 be the output generated using the smallest size linear operator (e.g. L_3 for first order operator). Using the pixel reference system shown in Fig. 20, a linear combination of the original edge map M_1 is used to generate an edge map at any level. Consider the generation of an edge map equivalent to the result of applying an L_5 sized operator to a hexagonal pixel-based image. The level 2 output, denoted as M_2, is computed directly from the level 1 edge map M_1. To obtain a gradient response at any pixel (x,y) in M_2, using the radial coordinate system in Fig. 20, we compute the following:

$$M_2(0,0) = M_1(0,0) + \frac{1}{2}\begin{pmatrix} M_1(1,0) + M_1(1,1) + M_1(1,2) \\ +M_1(1,3) + M_1(1,4) + M_1(1,5) \end{pmatrix} \quad (34)$$

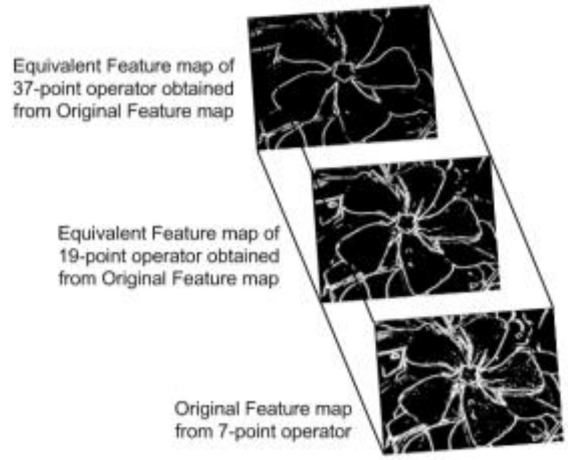


Fig. 22. Example of edge map scaling for the image shown in Fig. 17(e) to obtain resultant edge maps for various operator sizes

Stepping through each point in the edge map, M_2 can be readily obtained using this linear combination. This procedure can be generalised to compute edge maps at larger scales that correspond to convolution of the image with any operator size greater than the initial operator; in general an edge map at scale S (>1) can be constructed by a linear combination of values from the edge map at scale 1:

$$M_s = M_1(0,0) + \sum_{p=1}^{S-1} \sum_{q=0}^{6q-1} \left(\frac{S-p}{S}\right) M_1(p,q) \quad (35)$$

This procedure demonstrates a simplified approach to obtaining edge maps at multiple scales compared with conventional methods of constructing an operator at each scale and convolving each operator with the desired image. Instead, we need create only one 7-point hexagonal operator and apply this once to the image. It should be noted that this edge map scaling approach can be used to generate first order or Laplacian derivative operator edge maps at any scale.

As the output edge maps obtained from direct application of Linear-Linear operators are equivalent to those generated by the edge map scaling approach, it is not necessary to conduct comparative evaluation with respect to edge localisation when using the edge map scaling approach. However, in order to evaluate the efficiency, we provide run-times to determine increase in efficiency that can be achieved by using this approach.

We present the run-times for application of the Linear-Linear operator family to a hexagonal image, and the times taken to generate the equivalent edge maps using the edge map scaling approach. The results are provided in Table III, where the run-times (in milliseconds) are averaged over 100 runs on a PC with processor speed 2.88Ghz.

The results show that the generation of edge maps using the edge map scaling approach is more efficient than applying Linear-Linear scaled operators directly to hexagonal images. Using the proposed approach, it takes approximately half the time to generate the equivalent edge maps at each scale compared with those obtained by direct application of Linear-Linear operators.

TABLE III
RUN-TIMES TO COMPARE DIRECT OPERATOR APPROACH WITH EDGE MAP SCALING APPROACH

OPERATOR SIZE	$i = 3$	$i = 5$	$i = 7$
Application of Linear-Linear L_i operator to hexagonal image	5.64	10.45	17.72
Edge Map Scaling Approach run-time	5.64	6.06	9.10

VI. CONCLUSION

We have presented an approach to image processing operator construction that incorporates operator scalability using linear test functions within the finite element framework. In developing a linear hexagonal derivative operator the test functions used in the weak form are from the same space as those used in the image representation. Construction of first order and Laplacian Linear-Linear operators was demonstrated in Section 3.

We have demonstrated the efficient implementation of Linear-Linear operators through a process in which larger operators are generated using combinations of the 7-point Linear-Linear operator. This provides a more efficient way of constructing hexagonal operators at different scales than the conventional method of obtaining an operator by computing each element in the operator's neighbourhood and using finite element assembly to construct the operator, particularly as the size of the operator neighbourhood increases. Quantitative and qualitative methods were used to evaluate the accuracy of the proposed operators and the results obtained demonstrate that the Linear-Linear operators are comparable with the previously developed Linear-Gaussian operators [6].

We have presented an approach that provides an efficient method of obtaining scaled edge map outputs by directly using linear combinations of edge map outputs obtained at the lowest scale. The results show that when using the edge map scaling approach, significant computational gain is achieved with no reduction in the accuracy of the detected edges thus demonstrating the benefits of the proposed family of scalable derivative operators for edge detection tasks.

Furthermore this approach forms a framework for edge detection within the context of Scale Space Theory, where combinations of derivatives at various levels in zero-crossing based edge detection algorithms have been used. In the work on edge detection with automatic scale selection developed in [40], significant scale-space edge points are identified by maxima of specified edge strength measures that are located by zero-crossings of two functions (of varying order) of the scale-space image. This approach can be very successful in both identifying the most significant edges and demonstrating how the most salient scale varies along an edge. In [41] we have developed an alternative approach that naturally and systematically combines the smoothing and discrete derivative approximation steps that are carried out separately in [40], thus avoiding the use of ad hoc finite difference approximations. This work can naturally be extended for use to hexagonal pixel based images via the proposed multiscale framework. Although discrete second derivative operators do not usually form the sole basis of edge detection methods, developments in the field of Scale Space theory have used combinations of derivatives at various levels in zero-crossing based edge detection algorithms [40] .

REFERENCES

[1] Burdescu, Dumitru Dan, et al. "Computational Complexity Analysis of the Graph Extraction Algorithm for 3D Segmentation." Services (SERVICES), 2014 IEEE World Congress on. IEEE, 2014.
[2] Burdescu, Dumitru, et al. "New algorithm for segmentation of images represented as hypergraph hexagonal-grid", Pattern Recognition and Image Analysis. Springer Berlin Heidelberg, pp. 395-402, 2011.
[3] Coleman, S.A. Gardiner, B., and Scotney, B.W., "Design of Feature Extraction Operators for use on Biologically Motivated Hexagonal Image Structures", Proceedings of the IAPR Conference on Machine Vision and Applications, Japan. IEEE Computer Society, pp. 178-181, 2009.
[4] Coleman, S.A., Gardiner, B. and Scotney, B.W., "Coarse Scale Feature Extraction Using the Spiral Architecture Structure", In: Intenational Conference on Pattern Recognition, Turkey. IAPR, pp. 2370-2373, 2012.
[5] Coleman, S.A., Scotney, B.W. and Suganthan, S., "Multi-scale Edge Detection on Range and Intensity Images", Pattern Recognition, 44(4), pp. 821-838, 2011.
[6] Coleman, S.A., Scotney, B.W., Gardiner, B., "Efficient Laplacian feature map pyramids in a hexagonal framework", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1466-1469, 2010.
[7] Coleman, Sonya, Bryan Scotney, and Bryan Gardiner. "Processing Hexagonal Images in a Virtual Environment." Image Analysis and Processing–ICIAP, Springer Berlin Heidelberg, pp. 920-928, 2009.
[8] Davies E.R., "Circularity – A New Design Principle Underlying the Design of Accurate Edge Orientation Filters", Image and Vision Computing 5, pp. 134-142, 1984.
[9] Davies, E., "Optimising computation of hexagonal differential gradient edge detector", Electronics Letters 27, pp. 1526-1527, 1991.
[10] Diplaxmi R. Waghule, R. S. Ochawar; "Overview on Edge Detection Methods", Electronic Systems, Signal Processing and Computing Technologies (ICESC), pp. 151-155, 2014.
[11] Gardiner, B., Coleman, S.A. and Scotney, B.W., "A Design Procedure for Gradient Operators on Hexagonal Images", International Machine Vision and Image Processing Conference (IMVIP 2007), Maynooth, Ireland. IEEE Computer Society, pp. 47-54, 2007.
[12] Gardiner, B., Coleman, S.A. and Scotney, B.W., "Comparing Hexagonal Image Resampling Techniques with Respect to Feature Extraction", In: 14th International Machine Vision and Image Processing Conference, University of Limerick. Cambridge Scholars, pp. 102-115, 2011.
[13] Gardiner, B., Coleman, S.A., & Scotney, B.W., "Multi-Scale Feature Extraction in a Sub- Pixel Virtual Hexagonal Environment", Irish Machine Vision & Image Processing Conference, pp. 47-54, 2008.
[14] He, X., & Jia, W., "Hexagonal Structure for Intelligent Vision", Information and Communication Technologies, First International Conference, pp. 52-64, 2005.
[15] He, X., Hintz, T., Wu, Q., & Jia, W., "A New Simulation of Spiral Architecture", Proc. of Int. Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, pp. 570-575, 2006.
[16] He, X., Jia, X., Li, J., Wu, Q., Hintz, T., "An Approach to Edge Detection on a Virtual Hexagonal Structure", Digital Image Computing Techniques and Applications, pp. 340-345, 2007.
[17] Heath, M., Sarkar, S., Sanocki, T., & Bowyer, K., "A Robust Visual Method for Assessing the Relative Performance of Edge-detection Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 (12), pp. 1338-1359, 1997.
[18] Hofmann, Peter, and Dirk Tiede. "Image segmentation based on hexagonal sampling grids." South-Eastern European Journal of Earth Observation and Geomatics 3, pp. 173-177, 2014.
[19] Huang, C., & Lin, C., "Bio-inspired Computer Fovea Model based on Hexagonal-type Cellular Neural Network", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 54 (1), pp. 35-47, 2007.
[20] Iyengar, S., & Deng, W., "An Efficient Edge detection Algorithm using Relaxation Labelling Technique", Pattern recognition, Vol. 28 (4), pp. 519-536, 1995.

[21] Jiang, Q., "Orthogonal and Biorthogonal FIR Hexagonal Filter Banks with Sixfold Symmetry", IEEE Transactions on Signal Processing, Vol. 56, pp. 5861-5873, 2008.
[22] Kerr, D., Coleman, S.A. and Scotney, B.W., "FESID: Finite Element Scale Invariant Detector", In: 15th International Conference on Image Analysis and Processing (ICIAP), Vietrisul Mare, Italy, pp. 72-81, 2009.
[23] Kim, J., & Kim, H., "Multiresolution-based Watersheds for Efficient Image Segmentation", Pattern recognition letters, Vol. 24 (1-3), pp. 473-488, 2003.
[24] Knaup, M., Steckmann, S., Bockenbach, O., &Kachelries, M.,"CT Image Reconstruction using Hexagonal Grids", IEEE Medical Imaging Conference Rec , Vol. 13, pp. 2074-3076, 2007.
[25] Lin Wang, Xiangjian He, Ruo Du, Wenjing Jia, Qiang Wu, Wei-Chang Yeh, "Facial Expression Recognition on Hexagonal Structure Using LBP-Based Histogram Variances", MMM (2), pp. 35-45, 2011.
[26] Lindeberg, T., "Edge Detection and Ridge Detection with Automatic Scale Selection", International Journal of Computer Vision , Vol. 30 (2), pp. 117-156, 1998.
[27] Lindeberg, T., "Feature Detection with Automatic Scale Selection", Int Journal of Computer Vision , Vol. 30 (2), pp. 79-116, 1998.
[28] Middleton, L., & Sivaswamy, J.,"Hexagonal image processing: a practical approach", Springer-Verlag New York Inc, 2005.
[29] Middleton, L., Sivaswamy, J., "Edge Detection in a Hexagonal-Image Processing Framework", Image and Vision Computing 19, pp. 1071-1081, 2001.
[30] Pratt, W.K., Digital Image Processing. Wiley, John & Sons, 1991.
[31] Scotney B.W., Coleman S.A., "Improving Angular Error via Systematically Designed Near-circular Gaussian-based Feature Extraction Operators", Pattern Recognition, 40(5), pp. 1451-1465, 2007.
[32] Sharif, Muhammad, et al. "Face Detection and Recognition Through Hexagonal Image Processing", Sindh Univ. Res. Jour. (Sci. Ser.) 44.2, pp. 541-548, 2012.
[33] Shima, T., Saito, S., Nakajima, M., "Design and Evaluation of More Accurate Gradient Operators on Hexagonal Lattices", IEEE Trans. Pattern Analysis and Machine Intelligence 32(6), pp. 961-973, 2010.
[34] Staunton, R.C., "The Design of Hexagonal Sampling Structures for Image Digitisation and Their Use with Local Operators", Image and Vision Computing, pp. 162-166, 1989.
[35] Veni, S., and K. A. Narayanankutty. "Vision-based hexagonal image processing using Hex-Gabor", Signal, Image and Video Processing, 8.2, pp. 317-326, 2014.
[36] Wallace, R., Ong, P., Bederson, B., & Schwartz, E., "Space Variant Image Processing", International Journal of Computer Vision, Vol. 13 (1), pp. 71-90, 1994.
[37] Wu, Q., He, X., & Hintz, T., "Virtual Spiral Architecture", International Conference on Parallel and Distributed Processing Techniques and Applications", Vol. 1, pp. 399-405, 2004.
[38] Wüthrich, C.A., Stucki, P., "An algorithmic comparison between square- and hexagonal-based grids" CVGIP: Graphical Model and Image Processing 53(4), pp. 324-339, 1991.
[39] Xiangjian He; Wenjing Jia; Qiang Wu, "An approach of canny edge detection with virtual hexagonal image structure," Control, Automation, Robotics and Vision, ICARCV, pp. 879,882, 17-20 Dec. 2008.
[40] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection", Int. J. Computer Vision, 30 (2), pp. 117–156, 1998.
[41] B.W. Scotney, S.A. Coleman, M.G. Herron, "Device Space Design for Efficient Scale-Space Edge Detection", ICCS, LNCS 2329, Springer-Verlag, pp. 1077−1086, 2002.

Dr Bryan Gardiner received a BEng (Hons) in Electronics and Computer Systems from the University of Ulster, UK in 2006, and completed his PhD at Ulster in December 2010. He is a lecturer in the Intelligent Systems Research Centre at Ulster, and an active member in the Cognitive Robotics research group. Dr Gardiner's research interests are primarily in mobile robotics, digital image processing, computer vision and pattern recognition. He is a member of the IEEE UKRI society, International Association of Pattern Recognition and the Irish Pattern Recognition and Classification Society.

Professor Sonya Coleman received a BSc(Hons) in Mathematics, Statistics and Computing from the University of Ulster, UK in 1999, and a PhD in Mathematics from the University of Ulster, UK in 2003. She is the Cognitive Robotics team leader and currently Head of the Faculty Research Graduate School. She has 120+ publications primarily in the field of mathematical image processing, robotics and computational neuroscience. Prof. Coleman's research has been supported by funding from EPSRC award EP/C006283/11, the Nuffield Foundation, and the Leverhulme Trust. Additionally she was co-investigator on the EU FP7 funded project RUBICON and is currently involved in the FP7 projects VISUALISE and SLANDIAL. In 2009 was awarded the Distinguished Research Fellowship by the University of Ulster in recognition of her contribution research.

Professor Bryan Scotney received a BSc in Mathematics from the University of Durham, UK in 1980, and a PhD in Mathematics from the University of Reading, UK in 1985. He is Professor of Informatics at the University of Ulster, where, from 2005 to 2015, he was Director of the Computer Science Research Institute. He has over 260 publications, spanning a range of research interests in mathematical computation, especially in digital image processing and computer vision, pattern recognition and classification, statistical databases, reasoning under uncertainty, and applications to healthcare informatics, official statistics, biomedical and vision sciences, and telecommunications network management. In recent years he is an investigator on the EPSRC NETWORK in Next Generation Networks Systems and Services (EP/F030118/1), the EU FP7 Security programme projects SAVASA, on a Standards-based Approach to Video Archive Search and Analysis, and SLANDAIL, on a Security System for Language and Image Analysis. From 2007 to 2014 he was President of the Irish Pattern Recognition and Classification Society.