A Design Procedure for Gradient Operators on Hexagonal Images

Conference Paper · October 2007

DOI: 10.1109/IMVIP.2007.27 · Source: IEEE Xplore

CITATIONS
6

READS
165

3 authors:

B. Gardiner
Ulster University
27 PUBLICATIONS **70** CITATIONS
SEE PROFILE

Sonya Coleman
Ulster University
142 PUBLICATIONS **552** CITATIONS
SEE PROFILE

Bryan Scotney
Ulster University
258 PUBLICATIONS **1,741** CITATIONS
SEE PROFILE

A Design Procedure for Gradient Operators on Hexagonal Images

Bryan Gardiner[1], Sonya Coleman[1], and Bryan Scotney[2]
[1] *University of Ulster, Magee, BT48 7JL, Northern Ireland*
[2] *University of Ulster, Coleraine, BT52 1SA, Northern Ireland*
{gardiner-b, sa.coleman, bw.scotney}@ulster.ac.uk

Abstract

Image processing tasks have traditionally involved the use of square operators on regular rectangular image lattices. For many years the concept of using hexagonal pixels for image capture has been investigated, and several advantages of such an approach have been highlighted. Therefore, we present a design procedure for hexagonal gradient operators, developed within the finite element framework, for use on hexagonal pixel based images. In order to evaluate the approach we generate pseudo hexagonal images via resizing and resampling which also allows us to present results visually without the use of hexagonal lattice capture or display hardware. We provide comparative results with existing gradient operators, both rectangular and hexagonal.

1. Introduction

Conventionally, images are digitised and stored as a rectangular array of values. The image is sampled at each point on a two dimensional grid storing intensity and implicit location information for each sample. The rectangular grid is by far the most dominant of any grid structure in image processing although the hexagonal structured pixel grid is considered to be superior to the rectangular grid system in many respects, including greater angular resolution [8]. Use of hexagonal grids to represent digital images has been studied for more than 40 years [8, 17, 20], with recent improvements in charged coupled device (CCD) technology making hexagonal sampling attractive for practical applications and development of new interests in this area [10]. However, only recently have attempts been made to apply processing techniques directly to hexagonal images [10].

In machine vision, feature detection is often used to extract salient information from images. Image content often represents curved structures, which increase the complexity of the information being processed. With the use of most well known existing operators on a conventional rectangular lattice, limitations occur when detecting edges, most commonly due to methods on rectangular structures processing images principally in the horizontal and vertical directions. Potentially image information is excluded or lost due to failure to represent curves accurately [17]. One method to improve the treatment of curves objects is the use of compass operators to rotate feature detection masks to successfully detect diagonal edges [13]. Another approach would be to increase the image resolution if possible [11]. This does help to balance the loss of information, but the increase in image resolution generally leads to an increase in computational cost. To overcome this problem, an alternative sampling lattice, i.e. hexagonal, can be introduced.

There are a number of reasons why hexagonally structured images are considered beneficial. One of the major advantages is the consistency available in terms of neighbouring pixel distances when tiling an image plane. In a rectangular grid, the distance d between neighbouring pixel centres depends on whether the neighbour is vertically/horizontally adjacent, (with d=1), or diagonally adjacent (with d=$\sqrt{2}$) as illustrated in Figure 1(a).

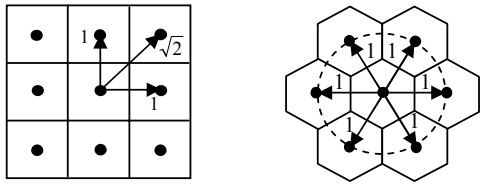


Figure 1. Pixel Lattices

In a hexagonal lattice, the distances between all neighbouring pixels are equal, i.e., d=1 in all cases, as shown in Figure 1(b). This creates a condition that will facilitate the implementation of circular symmetric kernels that is associated with an increase in accuracy when detecting edges, both straight and curved [2]. The accuracy of circular and near circular image processing operators has been demonstrated in [5, 6].

On a rectangular grid there are two possibilities that constitute a neighbourhood: the 4-neighbourhood structure, with horizontal and vertical neighbouring pixels only; or the 8-neighbourhood structure, which additionally uses the four diagonal pixels. On a hexagonal grid, neighbourhoods are simplified by consistently having six pixels in a neighbourhood, all equidistant from the centre pixel and sharing one edge and two corners with each adjacent pixel. Hexagonally shaped pixels are also advantageous as they utilize the

0-7695-2887-2/07 $25.00 © 2007 Crown Copyright
DOI 10.1109/IMVIP.2007.27

oblique effect in human vision [11]. A human viewing an image will be less sensitive to edges in diagonal directions compared with edges vertical and horizontal directions [10].

Sampling on a hexagonal grid has proven to achieve greater efficiency than on a square lattice and to incur less aliasing [9]. Vitulli [18] concluded that the sampling efficiency of hexagonal sampling exceeds that of square sampling, as approximately 13% less pixels are needed to obtain the same performance as obtained using square sampling. With 13% fewer pixels needed to represent a hexagonal image with the same information, less storage in memory will be needed for the image data, and hence less potentially computational time to process the image.

In recent work Coleman et al. [5] have focussed on the design and implementation of scalable and adaptive derivative operators through the use of a finite element framework, with results proving successful compared with well-known intensity image feature detection operators [4, 14]. In addition, the framework has also been used to design and implement novel near circular derivative operators [5, 15] that have been shown to improve edge orientation angular error. This paper builds on this finite element framework to facilitate hexagonally structured images. We present a gradient operator, developed using the finite element framework, for use directly on hexagonal grids. In order to obtain hexagonal pixel based images, regular rectangular images must be resampled to a hexagonal lattice, and techniques to do so are presented in Section 2. Sections 3 and 4 discuss how hexagonal pixels are represented in an image, giving a brief overview of how finite elements are used on a hexagonal grid. Section 5 shows how to compute an edge detection operator specifically for hexagonal images. Section 6 presents the techniques with which we compare our proposed technique, and in Section 7 presenting visual comparative results. Section 8 provides a summary and details of future work to be undertaken.

2. Resampling techniques

Although there are obvious benefits with hexagonal based images, one of the main reasons for the lack of use of the hexagonal lattice is the absence of availability of hardware: both sensors that enable the capture of hexagonal images and devices that enable their display. In order for research to advance in this area a resampling technique must be incorporated to enable the processing and display of hexagonal images using existing square lattice hardware.

To date, within the area of hexagonal image processing, conversion of, or resampling of, regular square tiled images to hexagonal tiled images is typically necessary. The main techniques used to simulate a hexagonal grid based on the use of an original square grid are those in [7, 12, 17, 19, 20].

Rosenfeld [12] resampled images by shifting every second row of square pixels by a half pixel, resulting in a brick wall effect. As the pixel shape is square, both the vertical and horizontal sampling distances are equal; however, the equidistant property of the hexagon is not achieved, as the distance to diagonal neighbours is $2/\sqrt{3}$. Staunton [17] resolved this issue by resampling the original image using rectangular pixels, with a horizontal length of 1 but a vertical length of $\sqrt{3}/2$. Combining this new pixel structure with the half pixel shift, the hexagonal property of equidistance between all neighbouring pixels is satisfied. Using rectangular pixels in this way also satisfies the angular resolution property which requires 60^0 between the six neighbouring pixels in the resampled image.

He [7] created an imitative hexagonal structure in which each hexagonal pixel consists of four square pixels with its intensity level taken as the average of the four square pixel values. Again every second row has a shift of half the width of the new sampled pixel, which is equivalent to the width of an original pixel. This resampled structure again encounters a loss in image resolution and also does not comply with the equidistance property needed for true hexagon representation. Wu [20] proposed taking a square grid image and converting it to a virtual spiral architecture. This architecture exists only for image processing calculations. This virtual structure is created in computer memory to allow processing algorithms to be implemented in a virtual space. The result of this processing is then mapped back to a rectangular architecture for display purposes. The main advantages of this resampling technique are the retention of image resolution and lack of distortion to the image. As this method is implemented on a virtual architecture, the isotropic properties of the hexagonal structure are fully satisfied. Wuthrich [19] proposed a method of creating a pseudo hexagonal pixel, known as a hyperpel from a cluster of square pixels. Although this does not create a perfect representation of a hexagon, it creates a hexagon-like shape that complies with the main hexagonal properties.

The resampling technique that we use is that of Wuthrich [19], an overview of which is provided by Figure 2. We have modified this technique slightly by representing each pixel by a $n \times n$ pixel block, as in [11], in order to create a sub-pixel effect to enable the sub-pixel clustering illustrated in Figure 3; this limits the loss of image resolution. Each pixel of the original image is represented by a 7×7 pixel block of equal intensity in the new image [11]. This creates a resized image of the same resolution as the original image with the ability to display each pixel as a group of $n \times n$ sub pixels.

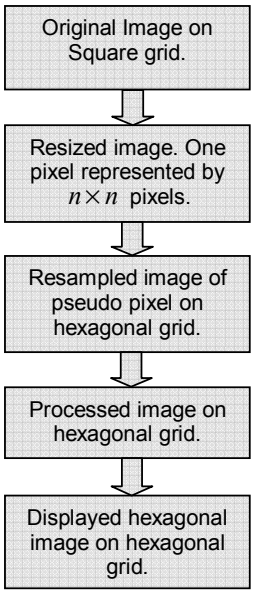


Figure 2. Resampling and processing on virtual grid

The motivation for image resizing is to enable the display of sub pixels, which is not otherwise possible. With this structure now in place, a cluster of sub pixels in the new image, closely representing the shape of a hexagon, can be created that represents a single hexagonal pixel in the resized image. Selection of the number of sub pixels to be clustered for each hexagonal pixel is based on two principle issues: the arrangement must allow a tessellation with no overlap and no gaps between neighbouring hexagonal pixels; and the cluster must closely resemble a hexagon i.e. six sides of approximately equal length. Two possible choices of hexagonal pixel representations are shown in Figure 3. Both cases represent a hexagonal pixel adequately with no overlap or gaps when tiling; however the cluster illustrated in Figure 3(b) is used as it is slightly superior in its ability to represent a hexagon in terms of having sides of near equal length [3].

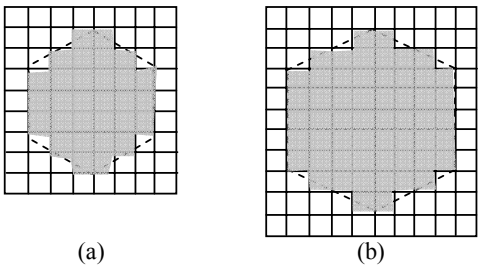


Figure 3. Two possible sub pixel clusters

Using this resizing and resampling enables us to display the resampled hexagonal image before and after image processing techniques have been applied.

3. Hexagonal image representation

With reference to the resized hexagonal image, it is possible to represent the image by using an array of samples of a continuous function $u(x, y)$ of image intensity on a domain Ω. Figure 4 represents hexagonal pixels with nodes placed in the centre of each pixel. These nodes are the reference points for the computation of finite element techniques throughout the domain Ω.

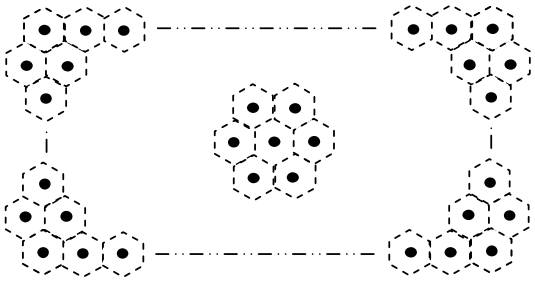


Figure 4. Representation of hexagonal image

4. Hexagonal operator design

We propose an operator for use on a hexagonal pixel based image as illustrated in Figure 4. The operator design procedure is based on the use of a "virtual mesh" illustrated in Figure 5, consisting of equilateral triangular elements, which overlays the pixel array shown in Figure 4.

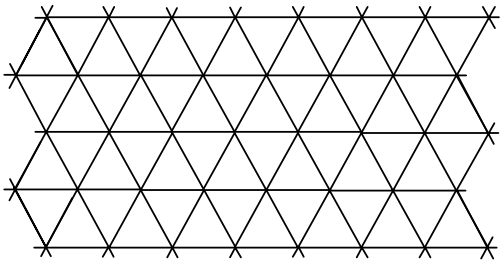


Figure 5. Virtual mesh of equilateral triangle elements

With any node in the virtual mesh, say node *i*, with co-ordinates (x_i, y_i) we associate a piecewise linear basis function $\phi_i(x, y)$ which has the properties

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

where (x_j, y_j) are the co-ordinates of the nodal point *j*. $\phi_i(x, y)$ is thus a "tent-shaped" function with support restricted to a small neighbourhood centred on node *i* consisting of only those triangular elements that have node *i* as a vertex. To construct an operator centred on node *i*, we define a neighbourhood Ω_i^{σ} consisting of a

compact subset of six elements. Denoting by D_i^σ the set of nodes contained in or on the border of Ω_i^σ, we may approximately represent the image u over the neighbourhood Ω_i^σ by a function

$$U(x,y) = \sum_{j \in D_i^\sigma} U_j \phi_j(x,y) \tag{2}$$

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 element in the neighbourhood Ω_i^σ and having intensity values $\{U_j\}$ at nodes $j \in D_i^\sigma$.

As in [4, 5, 16], we formulate operators that correspond to weak forms of operators in the finite element method [4, 16]. Operators used for smoothing may be based simply on a weak form of the image function. In this case it is assumed that the image function $u \equiv u(x,y)$ belongs to the Hilbert space $H^0(\Omega)$; that is, the integral $\int u^2 d\Omega$ over Ω is finite.

Edge detection and enhancement operators are often based on first or second derivative approximations, for which it is necessary that the image function $u \equiv u(x,y)$ belongs to the Hilbert space $H^1(\Omega)$; i.e., the integral $\int (|\underline{\nabla} u|^2 + u^2) d\Omega$ over Ω is finite, where $\underline{\nabla} u$ is the vector $(\partial u/\partial x, \partial u/\partial y)^T$. We are currently concerned only with first order derivative operators and therefore to obtain a weak form of the first directional derivative $\partial u/\partial b \equiv \underline{b} \cdot \underline{\nabla} u$ the derivative term is multiplied by a test function $v \in H^1$, and the result is integrated on the image domain Ω to give

$$E(u) = \int_\Omega \underline{b} \cdot \underline{\nabla} u v \, d\Omega \tag{3}$$

where $\underline{b} = (\cos\theta, \sin\theta)$ is the unit direction vector. This enables us to design our hexagonal operator using either a Cartesian coordinate system or the three axes of symmetry of the hexagon. Our current operator design uses the Cartesian coordinate system as the three axes of symmetry introduces redundancy. However, the symmetric hexagonal coordinate system has advantages when applied to tasks such as rotation that involve a large degree of symmetry [10], and hence may be used in future work.

In the finite element method a finite-dimensional subspace $S^h \subset H^1$ is used for function approximation; in our design procedure S^h is defined by the virtual finite element mesh in Figure 5. Although we are not addressing the issue of scale at this stage, our general design procedure incorporates a finite-dimensional test space $T_\sigma^h \subset H^1$ that explicitly embodies a scale parameter σ, enabling scale to be addressed in future work. The test space T_σ^h comprises a set of Gaussian basis functions $\psi_i^\sigma(x,y)$, i=1,…, N of the form

$$\psi_i^\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}\right)} \tag{4}$$

Each test function $\psi_i^\sigma(x,y)$ is restricted to have support over the neighbourhood Ω_i^σ, centred on node i. In general the size of Ω_i^σ may be explicitly related to the scale parameter σ [7]. However, in the current work Ω_i^σ is a simple six element hexagonal neighbourhood as illustrated in Figure 6.

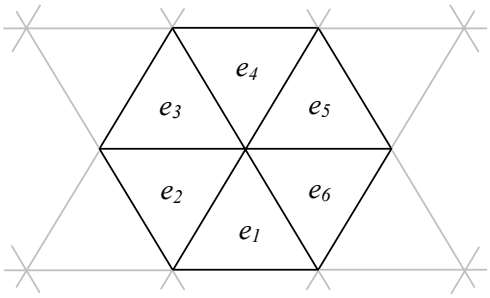


Figure 6. Hexagonal operator structure

The sets of test functions $\psi_i^\sigma(x,y)$, i=1, …, N, are then used in the weak forms of the first derivative in (3). In particular we note that the integrals need to be computed only over the neighbourhood Ω_i^σ, rather than the entire image domain Ω, since ψ_i^σ has support restricted to Ω_i^σ. Hence the approximate image representation over Ω_i^σ may be used, providing the functional

$$E_i^\sigma(U) = \int_{\Omega_i^\sigma} \underline{b}_i \cdot \underline{\nabla} U \psi_i^\sigma \, d\Omega_i \,. \tag{5}$$

5. Operator implementation

To illustrate the implementation of the first order hexagonal operator on a virtual mesh as shown in Figure 5, a general equilateral triangular element, as shown in Figure 7, will be used. Here one of the nodes α, β, λ, is a central node i of a neighbourhood Ω_i^σ. The neighbourhood Ω_i^σ covers a set of six elements $\{e_m\}$; where a Gaussian basis function ψ_i^σ is associated with the central node i which shares common support with the surrounding seven basis functions ϕ_j. Hence $E_i^\sigma(U)$ needs to be computed over the six elements in

the neighbourhood Ω_i^σ. Substituting the image representation in (2) into the functional $E_i^\sigma(U)$ in (5) yields

$$E_i^\sigma(U) = b_{i1}\sum_{j=1}^{N} K_{ij}^\sigma U_j + b_{i2}\sum_{j=1}^{N} L_{ij}^\sigma U_j \tag{6}$$

where $K_{ij}^\sigma = \sum_{m|e_m \in S_i^\sigma} k_{ij}^{m,\sigma}$ and $L_{ij}^\sigma = \sum_{m|e_m \in S_i^\sigma} l_{ij}^{m,\sigma}$ (7)

and $k_{ij}^{m,\sigma}$ and $l_{ij}^{m,\sigma}$ are the element integrals,

$$k_{ij}^{m,\sigma} = \int \frac{\partial \phi_j}{\partial x} \psi_i^\sigma \, dxdy \tag{8}$$

and

$$l_{ij}^{m,\sigma} = \int \frac{\partial \phi_j}{\partial y} \psi_i^\sigma \, dxdy \ . \tag{9}$$

In order to calculate $k_{ij}^{m,\sigma}$ and $l_{ij}^{m,\sigma}$, a local co-ordinate reference system for a equilateral triangle is introduced as illustrated in Figure 7, with co-ordinates ξ and η such that $\xi \geq 0$, $\eta \geq 0$ and $1-\xi-\eta \geq 0$. Mapping these global co-ordinates to local co-ordinates can be obtained by means of a co-ordinate transformation from e_m to $\hat{e}$ defined by (10) and (11).

$$x = (x_2^m - x_1^m)\xi + (x_3^m - x_1^m)\eta + x_1^m \, , \tag{10}$$

$$y = (y_2^m - y_1^m)\xi + (y_3^m - y_1^m)\eta + y_1^m \, . \tag{11}$$

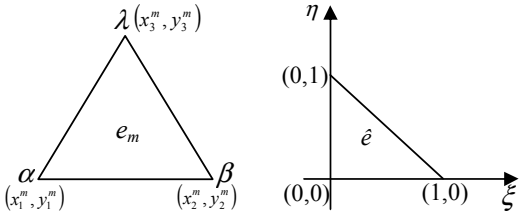


Figure 7. Local co-ordinate reference for equilateral triangle

It is required to compute the *x*- and *y*-directional derivative element matrices, K_i^e and L_i^e respectively, for each element e_m, as in [4]. Hence

$$K_i^e = \left[\left(y_2^m - y_3^m\right)\left(y_3^m - y_1^m\right)\left(y_1^m - y_2^m\right)\right]A_i \tag{12}$$

$$L_i^e = \left[\left(x_3^m - x_2^m\right)\left(x_1^m - x_3^m\right)\left(x_2^m - x_1^m\right)\right]A_i \tag{13}$$

where

$$A_i = \int_{\hat{e}} \psi_i^\sigma \, d\xi \, d\eta \qquad (i = 1,2,3) \tag{14}$$

The integration in (14) is achieved via the use of a four-point Gauss-Legendre numerical integration rule over the right-angled reference triangle.

With K_i^e and L_i^e calculated for each element in the neighbourhood, two masks (16) and (17) may be created by element assembly to compute the horizontal and vertical gradients respectively. Values for *a*, *b* and *c* are 0.22061, 0.43174 and 0.37665 respectively.

$$K = \begin{bmatrix} -a & a \\ -b & 0 & b \\ -a & a \end{bmatrix} \tag{16}$$

$$L = \begin{bmatrix} c & c \\ 0 & 0 & 0 \\ -c & -c \end{bmatrix} \tag{17}$$

The *L* operator can be seen to work in the same manner as a *y*-directional mask of a conventional 3×3 operator, e.g., Prewitt operator, being aligned along the vertical direction and detecting edges running perpendicular to the *y*-axis. As a hexagon naturally has three axes of symmetry, *x*, *y* and *z*, there are three possible edge masks that can be created to align with each of the axis directions simply by appropriate rotation of the *L* mask [10]. The *K* operator has no zeros running along a particular axis due to its orientation with respect to a hexagonal grid structure.

6. Feature detection techniques

The basic assumption used in most feature detection techniques is that edges in images are detected by a significant change in intensity of neighbouring pixels. Hence, the first derivative at the edge location will be at a maximum perpendicular to an edge. For comparison, we use three other gradient operators, Prewitt [11], the near circular Gaussian operator in [5] and the hexagonal Prewitt operator [10].

The Prewitt edge detector [11] is an example of a conventional rectangular gradient based detector. It approximates the gradient by the use of two 3×3 masks, aligned along both the horizontal and vertical axes. Generally, the Prewitt operator is considered to be a poor edge detector for standard images [10]. This is due to the fact that the masks compute a weak approximation to the gradient and fail to take into consideration the unequal inter-pixel distance in the 8-neighbour structure on which they operate. Despite these limitations, the Prewitt operator is still widely used due to its ease of implementation and low computational cost. The near circular (NC) operator [13] is a scalable operator that is based on the finite element framework. For fair comparison, we use only the regular rectangular 3×3 NC operator, again aligned along the Cartesian *x*- and *y*-axes.

The Prewitt hexagonal operator differs from the other operators as it is applied to a hexagonal lattice rather than a square lattice. The approach in Middleton [10] used to create these operators is based closely on the Prewitt square operator. The Prewitt hexagonal operator comprises three masks, rather than two, each aligned along one of the three axes of symmetry of the hexagonal lattice. The three masks are shown in equation (18)

$$h_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 \end{bmatrix} \qquad (18)$$

We can readily see the introduction of redundancy in (18) as $h_2 = h_1 - h_3$. Also, it should be noted that using this three axes framework, the gradient magnitude, *M*, is computed as

$$M = \sqrt{h_1^2 + h_3^2 + h_1 h_3} \qquad (19)$$

7. Experimental results

In order to compare the proposed hexagonal operator with the operators outlined in Section 6, three synthetic images containing a disc of radius 20, 50 and 90 pixels respectively, were used. Each image was represented using both a square and a hexagonal lattice to enable not only comparison of outputs from different hexagonal operators, but also to compare the output obtained from images using different pixel shapes. The edge maps are generated for each disc image, and for illustration purposes, slightly more than a quarter segment of each image is shown to enable close examination over all edge orientation. The images and corresponding edge maps are shown in Figures 8, 9 and 10 respectively.

In each figure, we see that the hexagonal image representation provides a smoother disc edge than the square pixel based image regardless of the disc's radius. Figure 8 illustrates that the hexagonal operators provide a more smoothly curved edge map, with the output from our proposed method (Figure 8(f)) being slightly better that the equivalent output from the Prewitt hexagonal operator (Figure 8(d)).

Figure 9 and Figure 10 present similar results to Figure 8 for the discs of radius 50 and 90 pixels. In both figures the hexagonal operators (Figures 9(d), 9(f), 10(d) and 10(f)) generated a more smoothly curved edge map than the corresponding regular square operators (Figures 9(c), 9(e), 10(c) and 10(e)).

From a visual comparison of the outputs from the

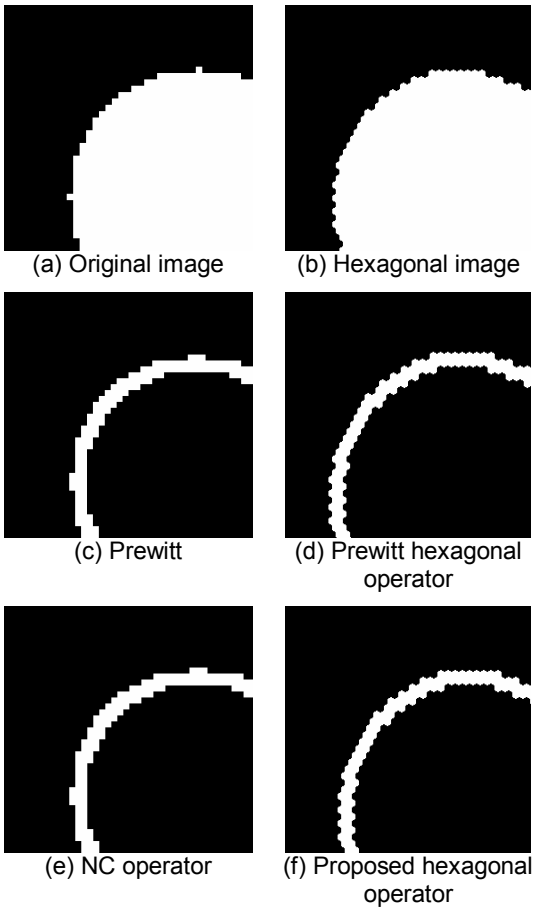


Figure 8. Edge detection results for disc of radius 20 pixels

hexagonal operators, we see that the proposed hexagonal operator provides a more smoothing curved edge map (Figures 9(f) and 10(f)) than the Prewitt hexagonal operator (Figures 9(d) and 10(d)); this is particularly apparent where the orientation of the disc edge is near vertical.

In order to compare our proposed hexagonal operator with the Prewitt hexagonal operator, we compute the Figure of Merit value [1] over a range of noise levels using synthetic curved edges rather than the standard synthetic horizontal and vertical edges. Figure 11 presents the Figure of Merit results for each operator and it can be seen that at each signal to noise ratio the proposed hexagonal operator detects the curved edge more accurately than the Prewitt operator.

For further comparison, each operator was applied to real images; both square pixel and hexagonal pixel based and outputs for Lena are shown in Figure 12.
The visual results presented are promising and demonstrate advantages of hexagonal images with respect to detecting curved edges. Additional analysis is required to provide quantitative evaluation of the use of such hexagonal images and operators in comparison

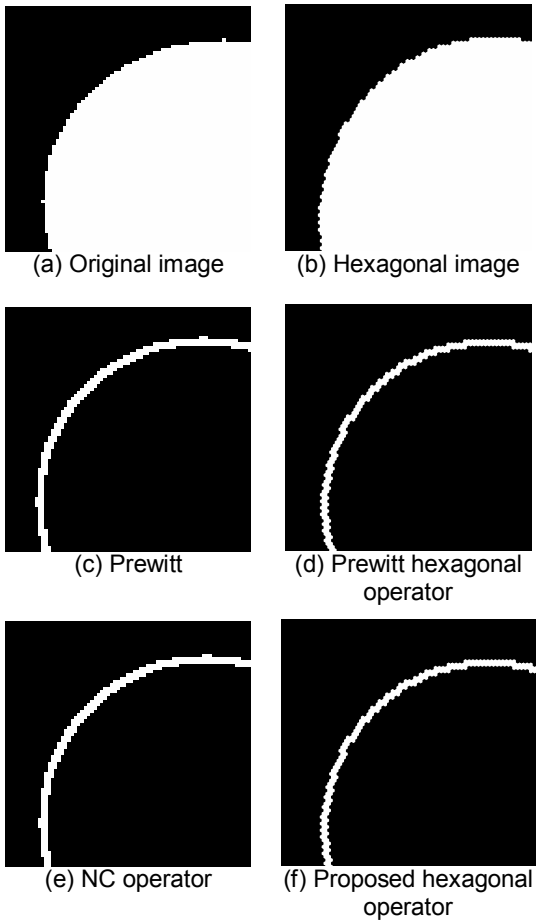


Figure 9. Edge detection results for disc of radius 50 pixels

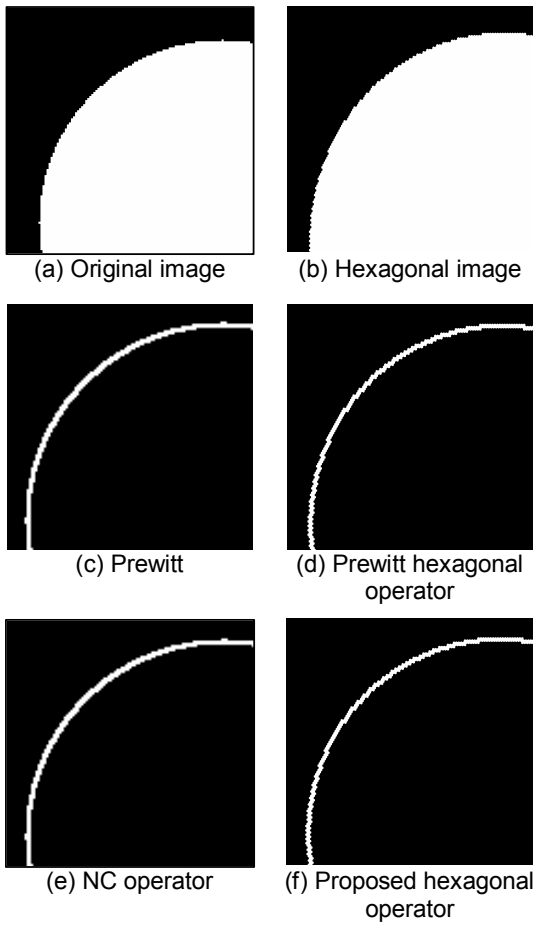


Figure 10. Edge detection results for disc of radius 90 pixels

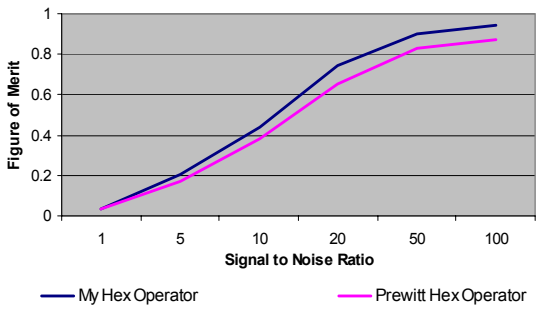


Figure 11. Figure of merit comparing both hexagonal operators

to the use of standard images and image processing operators.

8. Summary

We have presented a new hexagonal gradient operator for use on hexagonal pixel based images, created within the finite element framework. We have presented qualitative results that illustrate that the use of such hexagonal operators and images provides visual improvements with respect to detecting curved edges compared with the equivalent regular square gradient operators on a standard rectangular lattice. Further work will involve a thorough quantitative analysis for the proposed hexagonal operator, using straight and curved edges which will require the development of an extended Figure of Merit performance technique of curved edges. The hexagonal operator framework will also be extended to generate scalable hexagonal operators via the use of the virtual spiral architecture in [7].

9. References

[1] Abdou I. E., Pratt W. K., "Quantitative Design and Evaluation of Enhancement/ Threshold Edge Detectors." Proceedings of the IEEE, Vol. 67, No. 5, May 1979.

[2] Allen J.D., "Filter Banks for Images on Hexagonal Grid," Signal Solutions, 2003.

[3] Becker E.B., Carey G.F., Oden J.T., "Finite elements: An Introduction", Prentice Hall, London, 1981.

[4] Coleman, S.A., Scotney, B.W., Herron, M.G., "Content-Adaptive Feature Extraction Using Image Variance" *Pattern Recognition, Elsevier,* Vol. 38, pp2426-2436, 2005

[5] Coleman, S.A., Scotney, B.W. & Herron, M.G, "A Systematic Design Procedure for Scalable Near-Circular Laplacian of Gaussian Operators" *Proc. of Int. Conf. on Pattern Recognition (ICPR2004), Cambridge,* pp700-703.

[6] 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

[7] He X., "2-D Object Recognition with Spiral Architecture," University of Technology, Sydney, 1999.

[8] He X., Jia W., "Hexagonal Structure for Intelligent Vision," Information and Communication Technologies, ICICT, pp. 52- 64, 2005.

[9] Merseraue R.M., "The Processing of Hexagonally Sampled two-dimensional Signals," Proceedings of the IEEE 67, pp. 930-949, 1979.

[10] Middleton L., Sivaswamy J., "Hexagonal Image Processing; A Practical Approach," Springer, 2005.

[11] Middleton L., Sivaswamy, J., "Edge Detection in a Hexagonal-Image Processing Framework," Image and Vision Computing 19, pp. 1071-1081, June 2001.

[12] Rosenfeld, A., "Distance Functions on Digital Pictures," Pattern Recognition 1, pp. 33-61, 1968.

[13] Ruzon M.A., Tomasi, C., "Colour Edge Detection with the Compass Operator," Computer Vision and Pattern Recognition, IEEE Computer Society, Vol. 2, pp. 166-169, 1999.

[14] Scotney, B.W., Coleman S.A., Herron M. G., Sloot Peter M.A., Tan C.J. Kenneth, Dongarra Jack J, Hoekstra Alfons G., "Device Space Design for Efficient Scale-Space Edge Detection." Proc of Int. Conf. on Computational Science (ICCS). Springer-Verlag, LNCS, 2002.

[15] Scotney, B.W., Coleman, S.A., Herron, M.G., "Improving Angular Error by Near Circular Operator Design" *Pattern Recognition, Elsevier* Vol. 37(1), pp169-172, 2004.

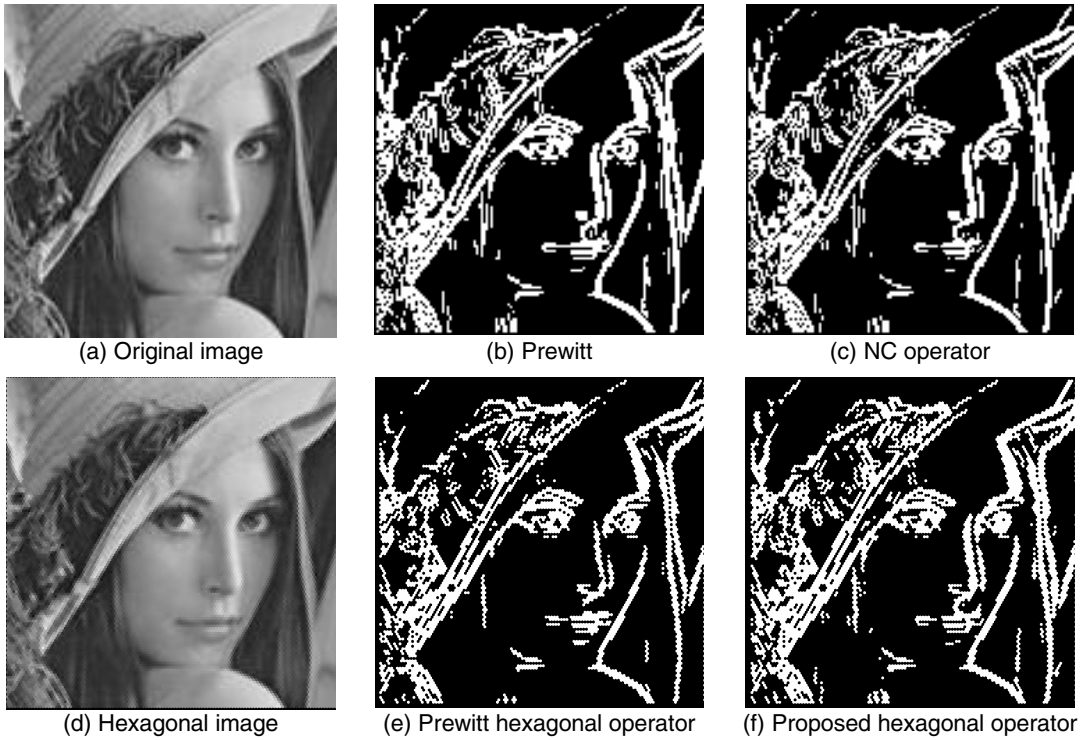
[16] Scotney B.W., and Herron M.G., "Systematic and Efficient Construction of Neighbourhood Operators on Irregular and Adaptive Grids," Proc. 3rd Irish Machine Vision and Image Processing Conference, pp. 204-218, 1999.

[17] Staunton R.C., "The design of hexagonal sampling structures for image digitisation and their use with local operators," Image Vision Computing, vol. 7, no. 3. pp 162-166, Aug 1989.

[18] Vitulli R., "Aliasing Effects Mitigation by Optimized Sampling Grids and Impact on Image Acquisition Chains," Geoscience and Remote Sensing Symposium, pp. 979-981, 2002.

[19] Wuthrich C.A., Stucki P., "An Algorithm Comparison between Square and Hexagonal Based Grids," CVGIP: Graphical Models and Image Processing 53, pp. 324-339, 1991.

[20] Wu Q, He X, Hintz T., "Virtual Spiral Architecture," International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 339-405, 2004.



(a) Original image (b) Prewitt (c) NC operator

(d) Hexagonal image (e) Prewitt hexagonal operator (f) Proposed hexagonal operator

Figure 12. Edge detection results - Lena image