Fast Multiscale Operator Development for Hexagonal Images

Conference Paper · September 2009

DOI: 10.1007/978-3-642-03798-6_29 · Source: DBLP

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

Fast Multiscale Operator Development For Hexagonal Images

[1]Bryan Gardiner, [1]Sonya Coleman, and [2]Bryan Scotney

[1]University of Ulster, Magee, BT48 7JL, Northern Ireland
[2]University of Ulster, Coleraine, BT52 1SA, Northern Ireland
gardiner-b@email.ulster.ac.uk
{sa.coleman, bw.scotney}@ulster.ac.uk

Abstract. 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. Recently there has been a renewed interested in using hexagonal pixel based images for various image processing tasks. Therefore, we present a design procedure for scalable hexagonal gradient operators, developed within the finite element framework, for use on hexagonal pixel based images. We highlight the efficiency of our approach, based on computing just one small neighbourhood operator and generating larger scale operators via linear additions of the small operator. We also demonstrate that scaled salient feature maps can be generated from one low level feature map without the need for application of larger operators.

1 Introduction

In machine vision, feature detection is often used to extract salient information from images. Image content often represents curved structures that may not be well represented on a rectangular lattice, and the characteristics of which may not be well captured by operators based on principal horizontal and vertical directions. The properties of operators developed on a rectangular grid are often influenced by the underlying Cartesian structure, i.e., operators may be dominated by the preferred directions along the x- and y-axes, leading to the inheritance of anisotropic properties. This problem may be further contributed to by the common practice of building operators using cross-products (in the co-ordinate directions) of existing one-dimensional operators. Such anisotropy is reflected in the spectral properties of the operators, and improvements can be achieved by developing operators that consider "circularity" [3, 4, 13]. One approach is the use of compass operators to rotate feature detection masks to successfully detect diagonal edges, whilst another is to increase image resolution if possible [13] to improve the representation of curved structures (but at increased computational cost). To overcome these problems, the hexagonal sampling lattice can be introduced, from which both spatial and spectral advantages may be derived: namely, equidistance of all pixel neighbours and improved spatial isotropy of spectral response. Pixel spatial equidistance facilitates the implementation of circular symmetric kernels that is associated with an increase in accuracy when detecting edges, both straight and curved [1], and the improved accuracy of circular and near circular image processing operators has been demonstrated in [4, 15]. Additionally, better spatial sampling efficiency is achieved by the hexagonal structure compared with a rectangular grid of similar pixel separation, leading to improved computational performance. In a hexagonal grid with unit separation of pixel centres, approximately 13% fewer pixels are required to represent the same image resolution as required on a rectangular grid with unit horizontal and vertical separation of pixel centres [18].

Due to the number of advantages offered by the hexagonal grid, the use of hexagonal lattices for image structure and representation has recently received renewed attention. The use of hexagonal pixel-based images dates back to the 1970s as the hexagonal structure is considered to be preferable to the standard rectangular structure typically used for images in terms of the improved accuracy and efficiency that can be achieved for a number of image processing tasks [5, 13]. More recently this area has become prominent with new developments in areas such as blue noise halftoning [12], hexagonal filter banks [10], image reconstruction [11, 19] and robot exploration [14] with applications including biologically inspired fovea modelling with neural networks that correspond to the hexagonal biological structure of photoreceptors [9], and the development of silicon retinas for robot vision [16, 17]. Although genuine hexagonal-based sensor systems and image capture devices do not yet exist, image representation in a hexagonal structure can be achieved readily through rectangular to hexagonal image conversion [8, 18, 19].

In [6, 7] we presented an approach to feature extraction operators using Gaussian test functions. In this paper, we present a novel and efficient approach to the design of hexagonal image processing operators using linear basis and test functions within the finite element framework. Section 2 describes the approach in [20] used to obtain hexagonal images with Section 3 and 4 presenting our efficient multi-scale operator design and algorithmic performance is presented in Section 6. Section 7 provides a summary and details of further work.

2 Hexagon images

To date, a hexagonal image can only be obtained by re-sampling a standard square pixel-based image. We have chosen to use the approach of [20] whereby hexagonal pixels are created through clusters of square sub-pixels. We have modified this technique slightly by representing each pixel by a pixel block, as in [13], in order to create a sub-pixel effect to enable the sub-pixel clustering; this modification limits the loss of image resolution. Each pixel of the original image is represented by a pixel block, Figure 1(a), of equal intensity in the new image [13]. This creates a resized image of the same resolution as the original image with the ability to display each pixel as a group of sub-pixels. 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, Figure 1(b).

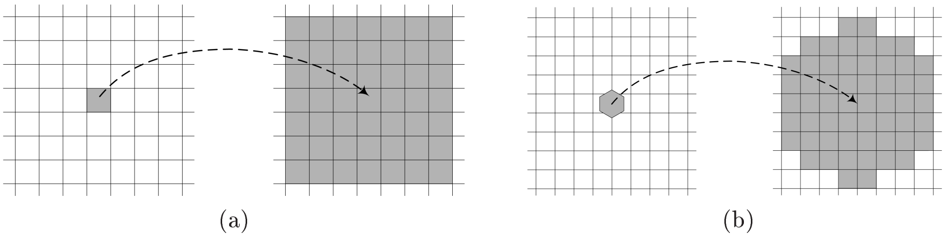


Fig. 1: Resizing of image to enable display of image at sub-pixel level

3 Hexagonal operator design

In order to develop scalable and efficient gradient operators for use on hexagonal image structures, we use the flexibility offered by the finite element framework. To achieve this, we initially represent the hexagonal image by an array of samples of a continuous function $u(x, y)$ of image intensity on a domain Ω with nodes placed in the centre of each pixel. These nodes are the reference points for finite element computation throughout the domain Ω, where the vertices of each triangular finite element are the pixel centres. Figure 2 represents an image compiled of hexagonal pixels with nodes placed in the centre of each pixel, overlaid by the triangular finite element mesh.

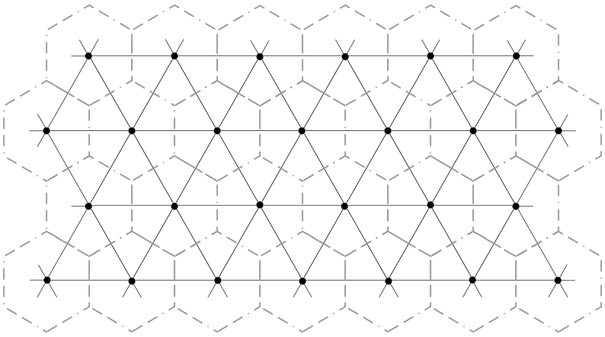


Fig. 2: Hexagonal array of pixels and overlying mesh

Given an image represented by an array of $n \times n$ samples of some continuous function $u(x, y)$ of image intensity, the goal here is to formulate operators involving a weak form of the directional derivative [2]. The weak form requires the image function to be once differentiable in the sense of belonging to the Hilbert space $H^1(\Omega)$. That is, $u = u(x,y)$ is such that the integral $\int\limits_{\Omega}(|\underline{\nabla}|^2 + u^2)d\Omega$ is finite, where Ω is the domain of the image, and $\underline{\nabla}u$ is the vector $(\delta u/\delta x,\ \delta u/\delta y)^T$. The derivative in the direction of a unit vector $\underline{b}$ may be expressed as $\delta u/\delta b$, where $\delta u/\delta b \equiv \underline{b} \cdot \underline{\nabla}u$. Thus by requiring that $u\epsilon H^1$, the problem is to find the weak form of the directional derivative of the image on the image domain Ω, namely

$$E(u) = \int\limits_{\Omega} \underline{b} \cdot \underline{\nabla}uvd\Omega \tag{1}$$

where v is a member of a function space H^1, and $\underline{b} = (\cos\theta, \sin\theta)$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 [13], and can be readily obtained from our Cartesian operators if required.

To approximate equation (1), a function in the image space H^1 may be approximately represented by a function from a finite-dimensional subspace $S^h \subset H^1$. The subspace S^h has a finite basis $\{\phi_1, ..., \phi_N\}$ and is of dimension N. So any function $V(x,y)\epsilon S^h$ may be uniquely represented by using a set of parameters $\{V_1, ..., V_N\}$ in the form

$$V(x,y) = \sum_{j=1}^{N} V_j \phi_j(x,y) \tag{2}$$

This form may be used to approximately represent the image u by a function $U\epsilon S^h$, where

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

and in which the parameters $\{U_1, ..., U_N\}$ are mapped from the sampled image intensity values. Using this representation of the image we may generate an approximate representation of the weak form of the directional derivative of the image by the functional

$$E_i(U) = \int\limits_{\Omega} \underline{b}_i \cdot \underline{\nabla}U\phi_i d\Omega \tag{4}$$

for each function ϕ_i in the basis of S^h. As the test functions ϕ_i used in the weak form are from the same space as those used in the approximate representation of the image, we may identify this formulation with the Galerkin method in finite element analysis.

We then construct a set of basis functions $\phi_i(x,y)$, $i = 1, ..., N$, so that the N-dimensional subspace S^h of H^1 comprises of functions which are piecewise polynomials. Such a basis for S^h may be formed by associating with each node i a basis function $\phi_i(x,y)$ which satisfies the properties

$$\phi_i(x_j,y_j) = \begin{cases} 1\,if & i=j \\ 0\,if & i\neq j \end{cases} \tag{5}$$

where (x_j,y_j) are the co-ordinates of the nodal point j in the finite element triangular mesh. Hence, $\phi_i(x,y)$ has a limited region of support Ω_i consisting of those elements which have node i as a vertex. The approximate image representation is thus a simple polynomial on each element and has the sampled intensity value U_j at node j, $j=1,...,N$.

4 Operator implementation

To illustrate the implementation of a first order hexagonal 3×3 operator we build a hexagonal operator as shown in Figure 3. The neighbourhood Ω_i covers a set of six elements em; 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. Hence $E_i^\sigma(U)$ needs to be computed over the six elements in the neighbourhood Ω_i.

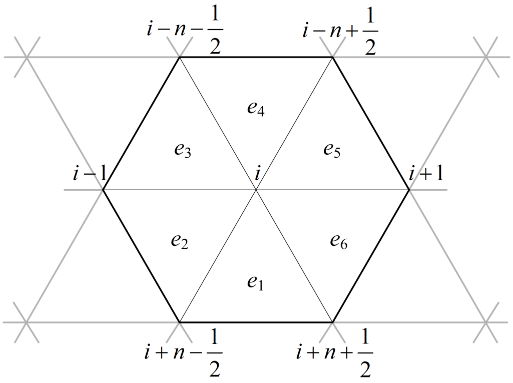


Fig. 3: Hexagonal operator structure

Substituting the image representation in (3) into the functional $E_i(U)$ in (4) yields

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

where

$$K_{ij} = \sum_{m|e_m \epsilon S^i} k_{ij}^m \quad and \quad L_{ij} = \sum_{m|e_m \epsilon S^i} l_{ij}^m \tag{7}$$

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

$$k_{ij}^m = \int \frac{\delta \phi_j}{\delta x} \phi_i dxdy \quad and \quad l_{ij}^m = \int \frac{\delta \phi_j}{\delta y} \phi_i dxdy \tag{8}$$

For each of the six triangular elements in the neighbourhood a pair of triangular element operators must be generated, whose entries then map directly to the corresponding locations within the 3×3 neighbourhood. For example, consider element e_1 shown in Figure 3. On this element the basis functions ϕ_j for $j = i$, $j = i + n - \frac{1}{2}$ and $j = i + n + \frac{1}{2}$ share common support with ϕ_i. Hence in this case the triangular element operators are

$$k_i^1 = \begin{bmatrix} & k_{i,i}^1 & \\ k_{i,i+n-\frac{1}{2}}^1 & & k_{i,i+n+\frac{1}{2}}^1 \end{bmatrix} \text{ and } l_i^1 = \begin{bmatrix} & l_{i,i}^1 & \\ l_{i,i+n-\frac{1}{2}}^1 & & l_{i,i+n+\frac{1}{2}}^1 \end{bmatrix} \tag{9}$$

where the entries in (9) are computed using the element integrals in (8) with the basis functions

$$\phi_i = \frac{y}{k} + 1; \qquad \phi_{i+n-\frac{1}{2}} = -\frac{x}{h} - \frac{y}{2k}; \qquad \phi_{i+n+\frac{1}{2}} = \frac{x}{h} - \frac{y}{2k} \tag{10}$$

Unlike with standard finite element matrices, the triangular element operators described here have two possible structures, those which are demonstrated in equation (9), suitable for elements e_1, e_3 and e_5, and those which corresponds to the structure of elements e_2, e_4 and e_6, for example:

$$k_i^2 = \begin{bmatrix} k_{i,i-1}^2 & & k_{i,i}^2 \\ & k_{i,i+n-\frac{1}{2}}^2 & \\ & & \end{bmatrix} \text{ and } l_i^2 = \begin{bmatrix} l_{i,i-1}^2 & & l_{i,i}^2 \\ & l_{i,i+n-\frac{1}{2}}^2 & \\ & & \end{bmatrix} \tag{11}$$

Similar triangular element operators are computed for each of the other elements in the neighbourhood and on completion, two masks (12) may be created by element assembly to compute the horizontal and vertical gradients respectively. Values for a, b and c are 0.1666, 0.3333 and 0.25 respectively.

$$K = \begin{bmatrix} & -a & & a & \\ -b & & 0 & & b \\ & -a & & a & \end{bmatrix} \quad \text{and} \quad \mathrm{L} = \begin{bmatrix} & c & & c & \\ 0 & & 0 & & 0 \\ & -c & & -c & \end{bmatrix} \tag{12}$$

5 Fast multiscale operator construction

On constructing the hexagonal equivalent of a 3×3 gradient operator, every other size of hexagonal operator (i.e., 5×5 ,7×7, etc) can be efficiently computed via the appropriate linear combinations of the 3×3 operator and this will be illustrated using the 5×5 operator as an example. Using the mesh in Figure 4(a) as a reference, in order to generate a 5×5 hexagonal operator, we place a 3×3 mask at the centre node of the mesh at level 0, node $(0, 0)$, and $\frac{1}{2} \times (3 \times 3)$ mask at the other six internal nodes at level 1. These are then combined in the typical finite element assembly manner as illustrated in Figure 4(b) for the 5×5 x-directional mask.

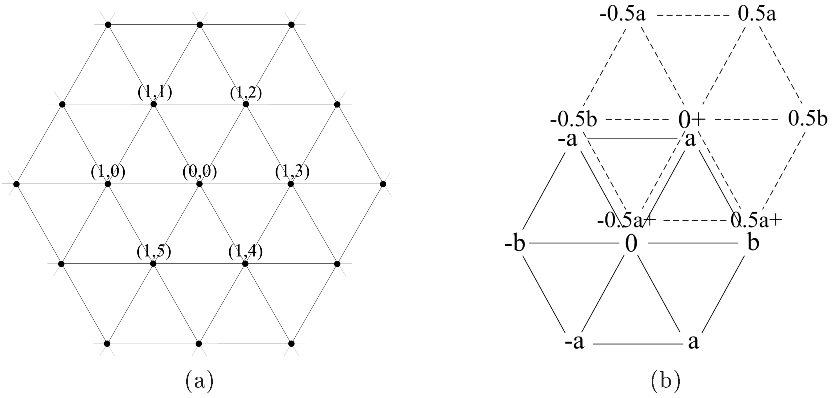


Fig. 4: (a) Finite element mesh corresponding to 2 neighbourhood levels, $i = 0, 1$ (b) Combining of the 3×3 masks to obtain the 5×5 mask, showing only one of the six masks for the second level

We can generalise this procedure for any operator size greater than 3 (the initial operator). Let the hexagonal operator size (5, 7 etc.) be denoted by S_o, then the radius of the approximately circular hexagonal operator R_o can be determined as

$$R_o = \frac{S_o - 1}{2} \tag{13}$$

Consider the nodes $(i,\ j)$ illustrated in Figure 4(a), here i indicates the level of the neighbourhood nodes, i.e., $i = 0$ at the centre node, $i = 1$ for each of the surrounding nodes at the next level etc., and j is each node within a given level i. For each operator size S (>3), the x-directional operator, S_{ox}, can be computed in an additive manner using the following formula:

$$S_{ox} = K_{(0,0)} + \sum_{i=1}^{R_o-1} \sum_{j=0}^{6i-1} \left(\frac{R_o - i}{R_o} \right) K_{(i,j)} \tag{14}$$

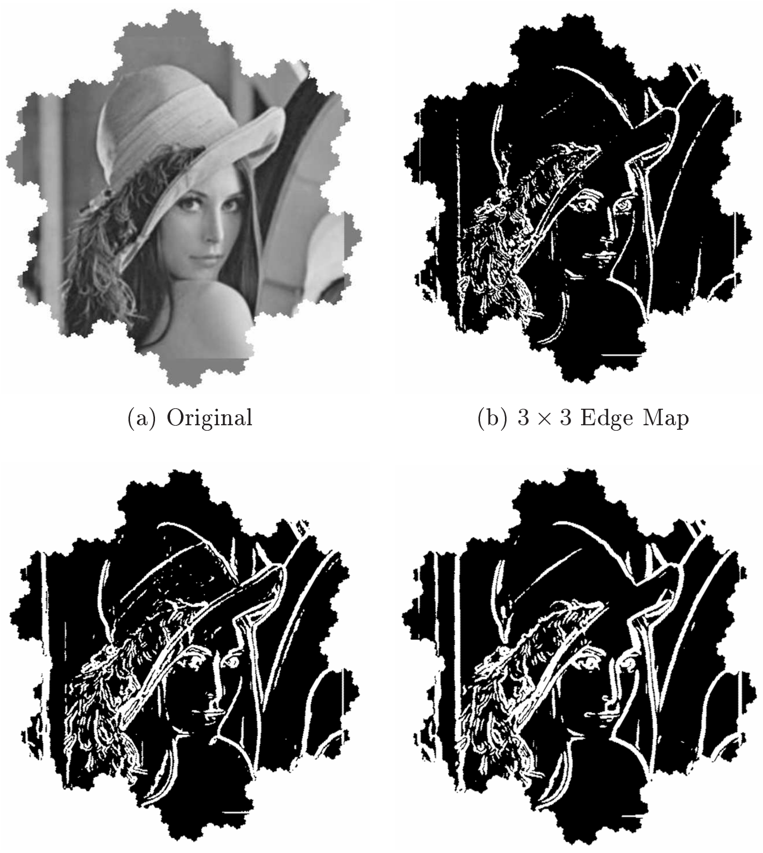
where $K_{(i,j)}$ is the 3×3 hexagonal mask placed at each node (i, j) and the number of levels, i, to be included is $R_o - 1$. Similarly, the y-directional operator, S_{oy}, can be computed as

$$S_{oy} = L_{(0,0)} + \sum_{i=1}^{R_o-1} \sum_{j=0}^{6i-1} \left(\frac{R_o - i}{R_o} \right) L_{(i,j)} \tag{15}$$

where $L_{(i,\ j)}$ is the 3×3 hexagonal mask placed at each node $(i,\ j)$.

6 Algorithmic performance

We illustrate the capability of the proposed operators by providing salient feature maps in Figure 5 generated by applying the operators to real images re-sampled onto a hexagonal pixel-based grid using the Lena image.



(a) Original

(b) 3 × 3 Edge Map

(c) 5 × 5 Edge Map

(d) 7 × 7 Edge Map

Fig. 5: Feature maps obtained using re-sampled real images

One of the advantages of hexagonal pixel-based images is that they contain 13.5% fewer pixels than a standard square pixel-based image of equivalent resolution. In addition, the hexagonal operators presented and designed on a Cartesian axis contain fewer operator values than the corresponding square operators, thus generating a significant overall reduction in computation. For example, for a given 256 × 256 image, removing boundary pixels, 63504 pixels will be processed.

Using a 5×5 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 feature map using an equivalent traditional square pixel-based image.

Operator	Run-time (ms)
Proposed Hexagonal	6.14
Prewitt	7.34
Sobel	7.11

Table 1: Run-times for 3×3 operators

We illustrate this further by providing run-times for the application of 3×3 operators to an image: the proposed 3×3 hexagonal operator to a hexagonal pixel-based image; and the 3×3 Prewitt and Sobel operators to a standard square pixel-based image. The run-times are presented in Table 1 and in each case the time is an average of 3 runs.

7 Summary

Recently, there has been a renewed interest in the use of hexagonal pixel-based images for image processing tasks as demonstrated in, for example, [8 - 13], however, much less research has been undertaken on the development and application of feature extraction operators for direct use on such image structures. Some standard algorithms have been extended from rectangular to hexagonal arrays in simple cases [5, 13], but generalisation and scalability of operators require a more systematic approach. We have presented a design procedure for scalable hexagonal operators developed for use within a finite element framework, the Galerkin formulation. We have demonstrated the efficient implementation of our approach through the efficient way in which larger operators can be generated using additions of the 3×3 operator. We have illustrated computation efficiently in Section 6, using an explicit example and runtimes for equivalent operators. Building on this promising performance, further research will involve evaluation of the hexagonal operators with respect to edge orientation and displacement, and extending the operators to interest point detectors and descriptors.

References

1. Allen J.D., “Filter Banks for Images on Hexagonal Grid,” Signal Solutions, 2003.
2. Becker E.B., Carey G.F., Oden J.T., “Finite elements: An Introduction”, Prentice Hall, London, 1981.

3. Coleman, S.A., Scotney, B.W. & Herron, M.G. "A Systematic Design Procedure for Scalable Near-Circular Laplacian of Gaussian Operators" Proceedings of the International Conference on Pattern Recognition, Cambridge, pp700-703, 2004
4. Davies, E.R., “Circularity - A New Design Principle Underlying the Design of Accurate Edge Orientation Filters”, Image and Vision Computing, Vol. 2, No. 3, pp. 134-142, 1984.
5. Davies, E.R., “Optimising Computation of Hexagonal Differential Gradient Edge Detector” Elect. Letters, 27(17) 1991
6. Gardiner, B., Coleman, S., & Scotney, B., “A Design Procedure for Gradient Operators on Hexagonal Images,” Irish Machine Vision & Image Processing Conference (IMVIP 2008),pp. 47-54, 2008.
7. Gardiner, B., Coleman, S., & Scotney, B., “Multi-Scale Feature Extraction in a Sub-Pixel Virtual Hexagonal Environment”, Irish Machine Vision & Image Processing Conference (IMVIP 2008),pp. 47-54, 2008.
8. He, X., Jia, W., “Hexagonal Structure for Intelligent Vision,” Information and Communication Technologies, ICICT, pp. 52- 64, 2005.
9. Huang, C-H., Lin, C-T., “Bio-Inspired Computer Fovea Model Based on Hexagonal-Type Cellular Neural Network” IEEE Trans Circuits and Systems, 54(1), pp35-47, Jan 2007
10. Jiang, Q., “Orthogonal and Biorthogonal FIR Hexagonal Filter Banks with Sixfold Symmetry” IEEE Transactions on Signal Processing, Vol. 56, No. 12, pp. 5861-5873, Dec 2008.
11. Knaup, M., Steckmann, S., Bockenbach, O., Kachelrieb, M., “CT Image Reconstruction using Hexagonal Grids” Proceedings of IEEE Nuclear Science Symposium Conference Record, pp. 3074-3076, 2007.
12. Lau, D. L., Ulichney, R., “Blue-Noise Halftoning for Hexagonal Grids” IEEE Transaction on Image Processing, Vol. 15, No. 5, pp. 1270-1284, 2006.
13. Middleton L., Sivaswamy J., “Hexagonal Image Processing; A Practical Approach,” Springer, 2005.
14. Quijano, H. J., Garrido, L., “Improving Cooperative Robot Exploration Using a Hexagonal World Representation” Proceeding of the 4th Congress of Electronics, Robotics and Automotive Mechanics, pp. 450-455, 2007.
15. Scotney, B.W., Coleman, S.A., "Improving Angular Error via Systematically Designed Near-Circular Gaussian-based Feature Extraction Operators" Pattern Recognition, Elsevier, Vol 40, No. 5, pp1451-1465, 2007.
16. Shimonomura, K., et al., “Neuromorphic binocular vision system for real-time disparity estimation” IEEE Int Conf on Robotics and Automation, pp.4867-4872, 2007.
17. Takami, R., et al., “An Image Pre-processing system Employing Neuromorphic 100 x 100 Pixel Silicon Retina” IEEE Int Symp Circuits & Systems, Vol. 3, pp2771-2774, 2005.
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. Wu, Q, He, X, Hintz, T., “Virtual Spiral Architecture,” Int Conf on Parallel and Distributed Processing Techniques and Applications, pp. 339-405, 2004.
20. Wuthrich, C.A, Stucki, P. “An Algorithmic Comparison Between Square-and Hexagonal-based Grid”, CVGIP: Graphical Models and Image Processing, Vol. 53, pp. 324-339, 1999.