Processing Hexagonal Images in a Virtual Environment

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

[1] School of Computing and Intelligent Systems, University of Ulster,
Magee, BT48 7JL, Northern Ireland
[2] School of Computing and Information Engineering, University of Ulster,
Coleraine, BT52 1SA, Northern Ireland

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 interest in the use of hexagonal images, representation of architectures for such images and general hexagonal image processing. Therefore, we present multiscale hexagonal gradient operators, developed within the finite element framework, for use directly on hexagonal pixel-based images. We demonstrate these operators using two environments: a *virtual hexagonal environment* and the direct use of simulated hexagonal pixel-based images. In both scenarios, we evaluate the proposed operators and compare them with the use of standard image processing operators on typical square pixel-based images, demonstrating improved results in the case of simulated hexagonal pixel-based images.

Keywords: Hexagonal pixel-based images, hexagonal gradient operators.

1 Introduction

Image processing tasks have traditionally involved the use of square operators on rectangular images lattices. A more recent concept that has been investigated is the use of hexagonal pixels for image capture, introducing the area of hexagonal image processing. Although hexagonal lattices have been explored for approximately 40 years [6, 12, 14], recent improvements in charged coupled device (CCD) technology have made hexagonal sampling attractive for practical applications and a renewed interest has been developed in this area [8, 9]. 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. There are a number of reasons why this is the case, and one major advantage is the consistency available in terms of neighbouring pixel distances when tiling an image plane, thus creating 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 [1]. Additionally, sampling on a hexagonal grid has proven to achieve greater efficiency than on a square lattice. Vitulli [13] concluded that the sampling efficiency of hexagonal sampling exceeds that of square sampling, as approximately 13% fewer pixels are needed to obtain the same performance as achieved 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 potentially less computational time to process the image. Although there are obvious benefits with hexagonal based images, image representation in a hexagonal structure can currently be achieved only through square to hexagonal image conversion [5, 12, 14, 16], though the emergence of genuine hexagonal-based sensor systems and image capture devices is crucial for the benefits of hexagonal structure to be fully appreciated and exploited.

Taking all this into consideration, few attempts have been made to process such hexagonal images. Although many feature detection operators exist for square lattices [2, 7], due to the structure of the hexagonal lattice these operators are not directly compatible with hexagonal pixel images. Only the simplest operators such as Sobel and Prewitt have, to date, been extended for use on hexagonal images [9, 12, 15], and only one gradient operator has been explicitly designed for use on hexagonally structured images [4]. However, more recently attempts have been renewed to apply processing techniques directly to hexagonal images, for example bi-lateral filtering based edge detection [15], introducing the possibility for significant development in the area of feature detection on these lattices.

In recent work [11] the finite element framework has been used to design and implement novel near-circular derivative operators that have been shown to improve edge orientation angular error. This paper builds on the finite element framework used in [11] to facilitate hexagonally structured images. We present an approach to the development of multiscale gradient operators explicitly 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 the procedure that we use to do so is described in Section 2. Sections 3 and 4 discuss how hexagonal pixels are represented in an image, giving a brief overview of how the gradient operators are implemented. In Section 5 we apply our hexagonal gradient operators in the *virtual hexagonal* environment for processing only, and then resample into a square pixel-based image for display and evaluation. Section 6 summarises the benefits obtained in the processing environment and discusses future work.

2   Creating a Hexagonal Pixel

To date, a hexagonal image can only be obtained by resampling a standard square pixel-based image. We have chosen to use the approach of [16] whereby hexagonal pixels are created through clusters of square sub-pixels. We have modified this technique slightly by representing each pixel by a $n \times n$ pixel block, as in [8], 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 7×7 pixel block, Fig. 1(a), of equal intensity in the new image [8]. 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. 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, Fig.1(b).
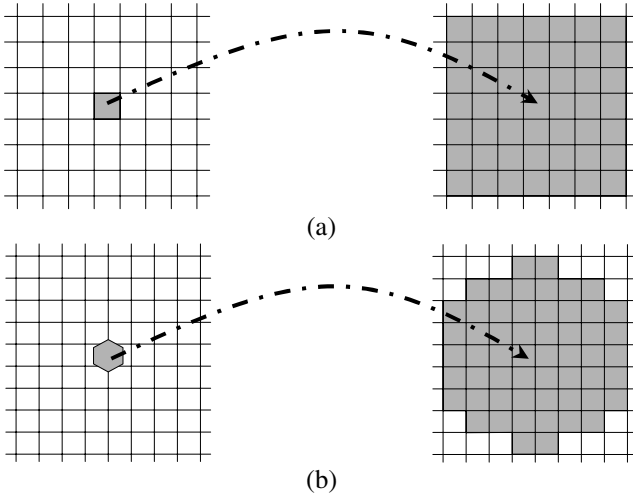
(a)

(b)

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

3   Representing a Hexagonal Pixel-Based Image

It is possible to represent the hexagonal image by using an array of samples of a continuous function $u(x, y)$ of image intensity on a domain Ω. Fig. 2 represents hexagonal pixels with nodes placed in the centre of each pixel. These nodes are the reference points for the computation of finite element based techniques throughout the domain Ω.
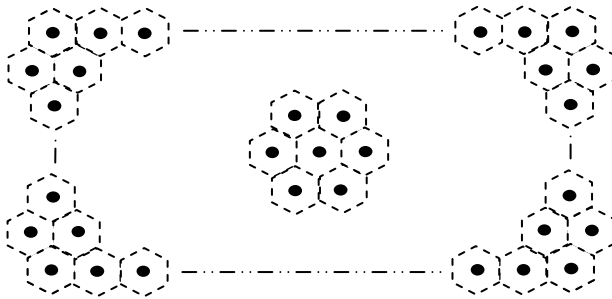

Fig. 2. Representation of a hexagonal image

4   Multiscale Hexagonal Operator Design

We propose an operator for use on a hexagonal pixel based image as illustrated in Fig. 2. The operator design procedure is based on the use of a "virtual mesh" illustrated in Fig. 3, consisting of equilateral triangular elements, which overlays the pixel array shown in Fig. 2.
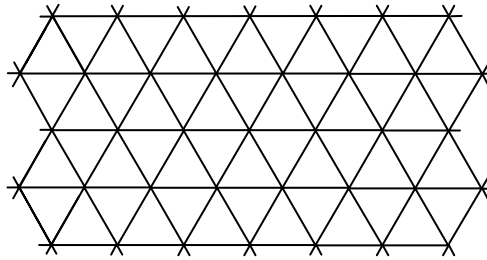
Fig. 3. Virtual mesh of equilateral triangular elements

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 $\phi_i(x_j, y_j) = 1$ if $i = j$ and $\phi_i(x_j, y_j) = 0$ if $i \neq j$ 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 $U(x, y) = \sum_{j \in D_i^\sigma} U_j \phi_j(x, y)$ 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 [11], we formulate operators that correspond to weak forms of operators in the finite element method. 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 $\int (|\nabla u|^2 + u^2) d\Omega$ over Ω is finite, where ∇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 $\partial u/\partial b \equiv \underline{b} \cdot \nabla u$ the derivative term is multiplied by a test function $v \in H^1$ and the result integrated over the image domain Ω to give

$$E(u) = \int_\Omega \underline{b} \cdot \nabla u v d\Omega \qquad (1)$$

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 [8]. However, the symmetric hexagonal coordinate system has advantages when applied to tasks such as rotation that involve a large degree of symmetry [8] and so 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 Fig. 3. Our general design procedure incorporates a finite-dimensional test space $T_\sigma^h \subset H^1$ that explicitly embodies a scale parameter σ and this test space T_σ^h comprises a set of Gaussian basis functions $\psi_i^\sigma(x, y)$, $i=1,\dots,N$. 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 [3], as illustrated by the six-element and 24-element hexagonal neighbourhoods in Fig. 4.
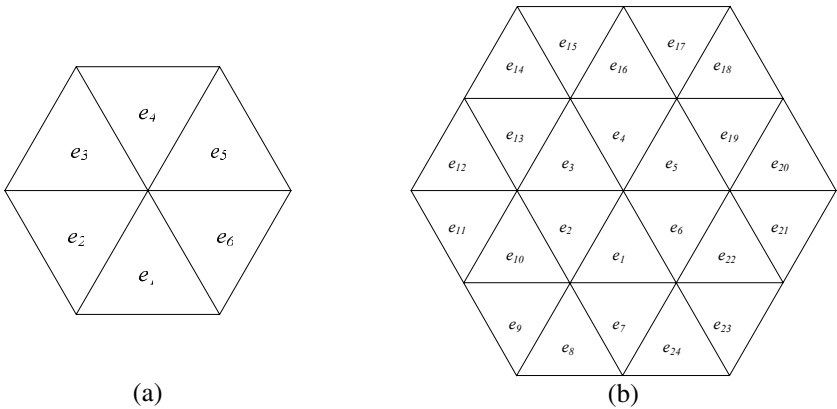

Fig. 4. Hexagonal operator structures: (a) six-element neighbourhood; (b) 24-element neighbourhood

The sets of test functions $\psi_i^\sigma(x, y)$, $i=1, \dots, N$, are then used in the weak forms of the first derivative in (2). 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

$$E_i^\sigma(U) = \int_{\Omega_i^\sigma} \underline{b}_i \cdot \nabla U \psi_i^\sigma \, d\Omega_i. \tag{2}$$

To illustrate the implementation of the first order hexagonal operator on a virtual mesh as shown in Fig. 3, a local co-ordinate reference system for a general equilateral triangular element will be used. Here one of the nodes α, β, λ, is a central node i of a neighbourhood Ω_i^σ. For example, the neighbourhood Ω_i^σ, in Fig. 4(a) covers a set of six elements $\{e_m\}$; 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 into the functional $E_i^\sigma(U)$ in (2) yields

$$E_i^\sigma(U) = b_{i1} \sum_{j=1}^{N} K_{ij}^\sigma U_j + b_{i2} \sum_{j=1}^{N} L_{ij}^\sigma U_j \tag{3}$$

where $K_{ij}^\sigma = \sum_{m|e_m \in S_i^\sigma} k_{ij}^{m,\sigma}$ and $L_{ij}^\sigma = \sum_{m|e_m \in S_i^\sigma} l_{ij}^{m,\sigma}$ and $k_{ij}^{m,\sigma}$ and $l_{ij}^{m,\sigma}$ are the element integrals $k_{ij}^{m,\sigma} = \int \frac{\partial \phi_j}{\partial x} \psi_i^\sigma \, dx dy$ and $l_{ij}^{m,\sigma} = \int \frac{\partial \phi_j}{\partial y} \psi_i^\sigma \, dx dy$. In order to calculate $k_{ij}^{m,\sigma}$ and $l_{ij}^{m,\sigma}$, we introduce 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 (4) and (5).

$$x = (x_2^m - x_1^m)\xi + (x_3^m - x_1^m)\eta + x_1^m, \tag{4}$$

$$y = (y_2^m - y_1^m)\xi + (y_3^m - y_1^m)\eta + y_1^m. \tag{5}$$

5 Processing in a Virtual Hexagonal Environment

In this section we present the results obtained when we resample a standard square pixel-based to a hexagonal pixel-based image, the *virtual hexagonal environment*, solely for the purpose of image processing and then resample back to the original square pixel-based image for display and evaluation as illustrated in Fig. 5. Within the *virtual hexagonal environment* feature extraction is performed using the multiscale hexagonal operators presented in Section 3.

The purpose of this is to determine whether, in the current absence of hexagonal image sensors, there is benefit to be gained by processing in a *virtual hexagonal environment*. Initially we provide Figure of Merit results in Fig. 6 using a ramp edge
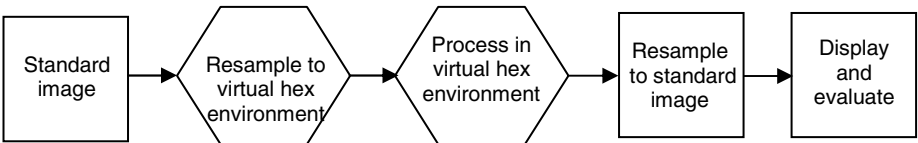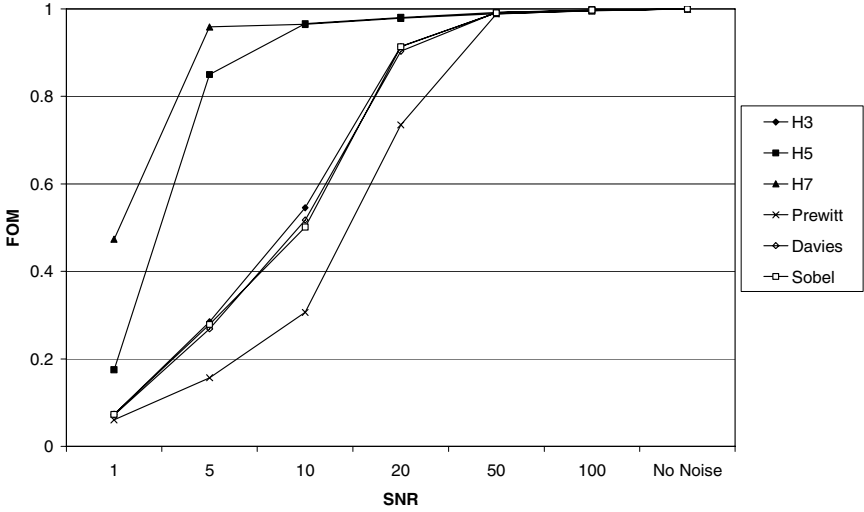

Fig. 5.

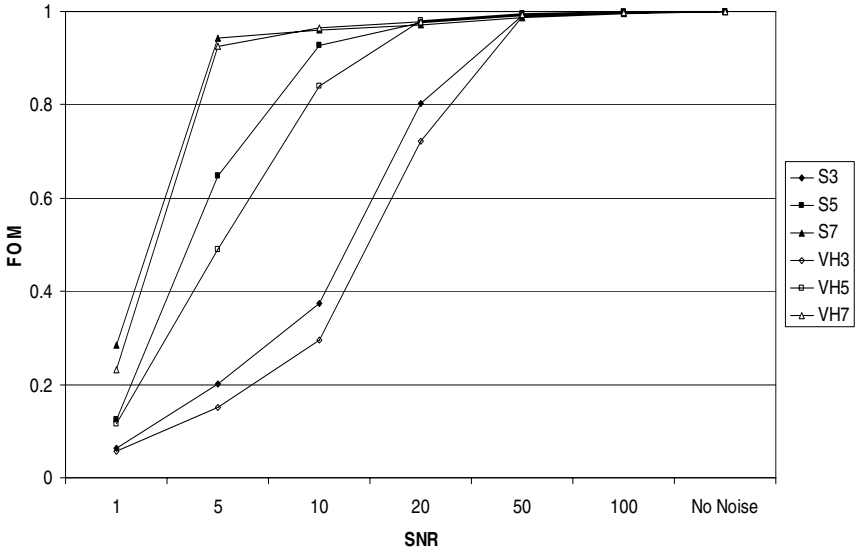**Fig. 6.** Figure or merit results using a 60° oriented edge

Fig. 7. Figure of merit results computed on a horizontal edge

oriented at 60° , comparing our proposed multiscale gradient operators (denoted H3, H5 and H7) with existing operators that can be directly used on hexagonal images, namely Davies [4], Prewitt and Sobel. The results show that our proposed approach provides improved edge localisation, over a range of signal-to-noise ratios, compared with the existing hexagonal operators.

In addition, we compute Figure of Merit results using the proposed hexagonal operators in the *virtual environment* (denoted VH3, VH5 and VH7) and the equivalent use of finite element based standard square operators [10] on standard images (denoted S3, S5 and S7). The results presented in Fig. 7 demonstrate that processing in a *virtual environment* provides results that are comparable to, and in some cases slightly better than, the equivalent use of typical operators on standard square pixel based images. We also illustrate real edge maps in Fig. 8, again illustrating that both approaches provide comparable results. These initial results are promising given that a hexagonal pixel-based image contains 13% fewer pixels than a standard square pixel-based image. Although computation is required to resample the original square pixel-based image to a hexagonal pixel-based image, the hexagonal operators 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 1587600. 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 1055754 multiplications, corresponding to 66.5% of the computation required to generate a similar feature map using an equivalent traditional square pixel-based image. The increase in accuracy when using our proposed hexagonal operators combined with the increase in efficiency when directly applying these operators will counteract the time taken to resample to a hexagonal grid.

(a) Original square pixel-based image

(b) Edge map using 5×5 standard square operator

(c) Edge map using 19-point hexagonal operator within the *virtual hex-environment*

Fig. 8. Edge maps obtained using Lena image

6 Conclusion

The use of hexagonal pixel-based images has received much attention in recent years with respect to both image architecture and processing. We have presented a design procedure for multi-scale gradient operators developed explicitly for use on hexagonal pixel-based images, created within the finite element framework. Using the Figure of Merit evaluation technique, we have demonstrated that the proposed operators can provide better feature localisation than currently existing operators for processing hexagonal images. In addition, we have demonstrated the use of such operators in a

virtual hexagonal environment, and the initial results obtained are promising. Therefore, we conclude that given the potential computational gain of using hexagonal images through both the fact that a hexagonal pixel-based image contains 13% fewer pixels than the equivalent square pixel-based image and the fact that hexagonal operators contain fewer operator values than their square-lattice counterparts, the use of hexagonal pixel-based images has potential be significantly beneficial to the computer vision and image processing communities. Future work will encompass the use of the spiral architecture in our operator implementation with the overall aim of providing an overall design and implementation framework that can reduce further the computation complexity of the approach. Hence the hexagonal image framework should become more appealing for use in real-time imaging applications.

References

[1] Allen, J.D.: Filter Banks for Images on Hexagonal Grid. Signal Solutions (2003)

[2] Canny, J.F.: A Computational Approach to Edge Detection. IEEE Trans on Pattern Analysis and Machine Intelligence 8(6) (November 1986)

[3] Davies, E.R.: Circularity – A New Design Principle Underlying the Design of Accurate Edge Orientation Filters. Image and Vision Computing 5, 134–142 (1984)

[4] Davies, E.R.: Optimising Computation of Hexagonal Differential Gradient Edge Detector. Electronic Letters 27(1)

[5] He, X.: 2-D Object Recognition with Spiral Architecture. University of Technology, Sydney (1999)

[6] He, X., Jia, W.: Hexagonal Structure for Intelligent Vision. In: Information and Communication Technologies, ICICT, pp. 52–64 (2005)

[7] Marr, D., Hildreth, E.: Theory of Edge Detection. In: Proceedings of the Royal Society of London, Series B, vol. 207, pp. 187–217 (1980)

[8] Middleton, L., Sivaswamy, J.: Hexagonal Image Processing; A Practical Approach. Springer, Heidelberg (2005)

[9] Middleton, L., Sivaswamy, J.: Edge Detection in a Hexagonal-Image Processing Framework. Image and Vision Computing 19, 1071–1081 (2001)

[10] Scotney, B.W., Coleman, S.A., Herron, M.G.: Device Space Design for Efficient Scale-Space Edge Detection. In: Sloot, P.M.A., Tan, C.J.K., Dongarra, J., Hoekstra, A.G. (eds.) ICCS-ComputSci 2002. LNCS, vol. 2329, pp. 1077–1086. Springer, Heidelberg (2002)

[11] Scotney, B.W., Coleman, S.A.: Improving Angular Error via Systematically Designed Near-Circular Gaussian-based Feature Extraction Operators. Pattern Recognition 40(5), 1451–1465

[12] Staunton, R.C.: The design of hexagonal sampling structures for image digitisation and their use with local operators. Image Vision Computing 7(3), 162–166 (1989)

[13] Vitulli, R.: Aliasing Effects Mitigation by Optimized Sampling Grids and Impact on Image Acquisition Chains. Geoscience and Remote Sensing Symposium, 979–981 (2002)

[14] Wu, Q., He, X., Hintz, T.: Virtual Spiral Architecture. In: Int. Conf. on Parallel and Distributed Processing Techniques and Applications, pp. 339–405 (2004)

[15] Wu, Q., He, X., Hintz, T.: Bi-lateral Filtering Based Edge Detection on Hexagonal Architecture. In: Proc. of Int. Conf. on Acoustic, Sound an Signal Processing, vol 2, pp. 713–716 (2005)

[16] Wuthrich, C.A., Stucki, P.: An Algorithm Comparison between Square and Hexagonal Based Grids. In: CVGIP: Graphical Models and Image Processing, vol. 53, pp. 324–339 (1991)