Integral Spiral Image for Fast Hexagonal Image Processing

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. A common requirement for image processing tasks is to achieve real-time performance. One approach towards achieving this for tradition rectangular pixel-based images is to use an integral image that enables feature extraction at multiple scales in a fast and efficient manner. Alternative research has introduced the concept of hexagonal pixel-based images that closely mimic the human visual system: a real-time visual system. To enhance real time capability, we present a novel integral image for hexagonal pixel based images and associated multi-scale operator implementation that significantly accelerates the feature detection process. We demonstrate that the use of integral images enables significantly faster computation than the use of conventional spiral convolution or the use of neighbourhood address look-up tables.

1 Introduction

Motivated by the real-time processing capabilities of the human vision system, we consider the use of hexagonal pixel-based images in order to reduce computational effort when implementing low-level image processing algorithms. We consider the way in which humans capture visual information: a small region, the fovea, within the retina, contains photoreceptor cones that are arranged in a densely packed hexagonal structure. Correspondingly, we consider digital images in which the pixels are hexagonal. In [8] a spiral architecture is designed which enables a hexagonal pixel-based image to be stored as a one dimensional vector. This is a fundamental characteristic that can be utilized to target real-time processing.

One of the most popular ways of applying edge detection operators at multiple scales is through the use of image pyramids [4]. An image pyramid is constructed by first smoothing the image with an appropriate filter and then sub-sampling the smoothed image. This process is repeated a number of times on the subsequently generated images resulting in a set of increasingly smoothed images. Edge detection, for example, is then performed by applying a gradient-based operator such as the Sobel operator to each image in the pyramid. However one issue with this method is that it is difficult to relate features at higher levels of the image pyramid to those at lower levels of the pyramid due to the fact that the spatial locations of the detected features do not relate directly. An alternative method to applying edge detection operators at multiple scales is to use is to use a set of differently sized operators applied to

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 532–541, 2013.

the image, resulting in a set of edges detected at different scales. However, it is difficult to generalise operators such as the Sobel and Prewitt operators to allow the mask size to be altered to facilitate such an approach.

One approach to overcoming computational overheads when performing feature extraction is the use of integral images. Integral images provide a means of fast computation when using small convolution filters, as the processing time and number of operations required to compute any area of the integral image is independent of the size of the region. Viola and Jones [9] first used rectangular integral images to perform real-time object detection discussing the computation time performance improvements. Integral images are also a key aspect of the SURF detector [1] and have also been used for adaptive thresholding [2] and object detection [3]. Therefore we present a novel hexagonal integral image, based on the spiral architecture to enable fast feature extraction.

In this paper we design integral spiral images that provide a framework for obtaining feature maps efficiently over a range of coarse scales. The use of integral images results in coarse scale feature extraction. However, in applications such as robot navigation, it is appropriate to conduct coarse scale processing to generally determine a robot's location (i.e., if it is in the map), and to then proceed with fine scale processing as necessary to determine the exact location [10]. In Section 2 we outline the spiral architecture in [8] and present our novel integral spiral image, followed by the 7-point operator design in Section 3. In Section 4 we describe the framework for fast processing and present results in Section 5.

2 Spiral Images

2.1 Spiral Architecture

In the spiral architecture [8] the addressing scheme for the spiral image, denoted by S, originates at the centre of the image (pixel index 0) and spirals out using one-dimensional indexing. Figure 2 shows the spiral addressing scheme for the central portion of an image. Pixel 0 may be considered as a layer 0 cluster. Pixel 0, together with its six immediate neighbours indexed in a clockwise direction (pixels 1,…,6) then form a layer 1 cluster centred at pixel 0. This layer 1 cluster may then be combined with its six immediately neighbouring layer 1 clusters, the centres of which are indexed as 10, 20, 30, 40, 50 and 60, to form a layer 2 cluster centred at pixel 0 (as shown in Figure 1); the remaining pixels in each of these layer 1 clusters are indexed in a clockwise direction in the same fashion as the layer 1 cluster centred at 0, (e.g., for the layer 1 cluster centred at 30, the pixel indices are 30, 31, 32, 33, 34, 35 and 36). The entire spiral addressing scheme is generated by recursive use of the clusters; for example, seven layer 2 clusters are combined to form a layer 3 cluster. Ultimately the entire hexagonal image may be considered to be a layer L cluster centred at 0 comprising 7^L pixels.

An important advantage of the spiral addressing scheme is that any location in the image can be represented by a single co-ordinate value, and hence the spiral image

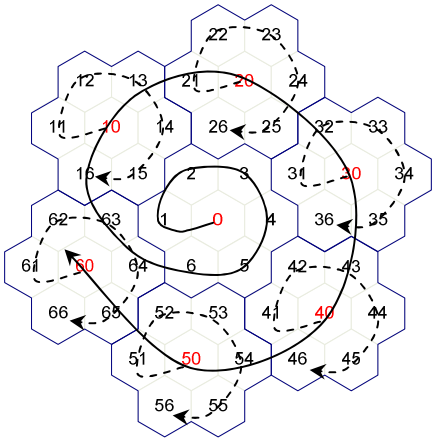

Fig. 1. One-dimensional addressing scheme in the central region of the image

can be stored as a vector [5]. Spatially neighbouring pixels within any 7-pixel layer 1 cluster in the image remain neighbouring pixels in the one-dimensional image storage structure. This is a very useful characteristic when performing image processing tasks on the stored image vector, and this contiguity property lies at the heart of our approach to achieve fast and efficient processing for feature extraction.

2.2 Integral Spiral Image

We introduce an integral spiral image, analogous to the traditional integral image approach in [9] for rectangular pixel-based images. As the spiral image S is represented by a vector, the integral spiral image, denoted by I, is computed in the following way:

$$I(p) = S(p) \quad \text{for pixel} \quad p = 0 \tag{1}$$

$$I(p) = S(p) + I(p-1) \quad \text{for pixel} \quad p \neq 0 \tag{2}$$

3 Operatori Design

The key aspect of the integral image approach is that only one 7-point operator is required that can be applied at multiple course scales (layers in the spiral architecture) using the integral spiral image presented in Section 2. To develop the 7-point operator we need to consider only Layer 1. To compute the operator components we use a regular mesh of equilateral triangles with nodes placed at the pixel centres (Figure 2(a)). With each node p we associate a piecewise linear basis function ϕ_p, with $\phi_p = 1$ at node p and $\phi_m = 0$ at all other nodes $m \neq p$. Each ϕ_p is thus a "tent-shaped" function with support restricted to a small neighbourhood of six triangular elements centred at node p (Figure 2(b)). We represent the spiral image by a

function $S = \sum_{q \in Q} S(q)\phi_q$, where Q denotes the set of all nodal addresses; the parameters $\{S(q)\}$ are the image intensity values at the pixel centres.

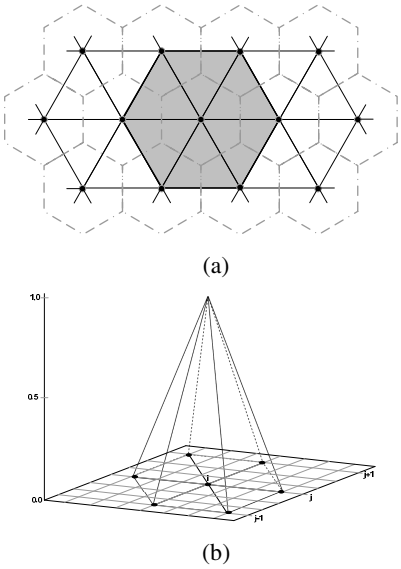

Fig. 2. (a) regular mesh of equilateral triangles with nodes placed at the pixel centres; (b) "tent-shaped" function

Feature detection and enhancement operators are often based on first derivative approximations, and we consider a weak form of the first directional derivative $\partial S/\partial b \equiv \underline{b} \cdot \underline{\nabla} S$. To approximate the derivatoive over a layer 1 cluster centred on the pixel with spiral address *p*, the image derivative is multiplied by a neighbourhood test function ψ_p and the result integrated over a neighbourhood $N_1(p)$ corresponding to the layer 1 cluster centred on pixel *p*. Hence at pixel *p* we obtain a directional derivative $D_1(p)$ in any direction $\underline{b}$ ($\underline{b}$ is a unit direction vector) as

$$D_1(p) = \int_{N_1(p)} \underline{b} \cdot \underline{\nabla} S \psi_p \, d\Omega \tag{3}$$

Thus we may write

$$D_1(p) = \sum_{q \in Q} \left(S(p) \int_{N_1(p)} \underline{b} \cdot \underline{\nabla} \phi_q \psi_p^1 \, d\Omega \right) = \sum_{q \in N_1(p)} H_1(q) \times S(q) \tag{4}$$

where H_1 is the 7-point Layer 1 hexagonal operator. We have chosen the neighbourhood test function ψ_p to be a Gaussian function restricted to $N_1(p)$, centred on node *p* and parameterised so that 95% of its central cross section falls within $N_1(p)$. The operator H_1, shown in Figure 3, is then used as the 7-point operator.

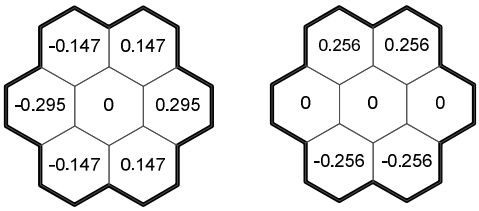

Fig. 3. x- and *y*-components of operator H_1

4 Operator Convolution

Typically feature extraction is achieved by convolving an operator at every location in the image providing a gradient magnitude value each pixel location and this is true when we apply the 7-point operator to every point in the hexagonal pixel-based image. However, when integral images are used this is not necessarily the case. To implement the operator H_1 using an integral image, we need to determine the cluster integrals $CI(c_i)$ for the seven layer (λ-1) clusters that comprise the layer λ cluster. Here, the values of c_i denote the centres of these seven layer as $c_i = s_0 + i10^{\lambda-1}$ for $i = 0,...,6$. Using base 7 addition [7], the layer (λ-1) cluster integral value at c_i is then calculated as (λ-1) clusters. For a layer λ cluster with centre s_0, the seven corresponding layer (λ-1) cluster centres are computed as $c_i = s_0 + i10^{\lambda-1}$ for $i = 0,...,6$. Using base 7 addition [7], the layer (λ-1) cluster integral value at c_i is then calculated as

$$CI(c_i) = I(6\sum_{k=0}^{\lambda-1} 10^k) \quad \text{for } c_i = 0 \tag{5}$$

$$CI(c_i) = I(c_i + 6\sum_{k=0}^{\lambda-1} 10^k) - I(c_i - 1) \quad \text{for } c_i \neq 0 \tag{6}$$

The operator at scale λ, applied to a layer λ cluster, is implemented by convolving a core 7-point operator with the cluster integral values $CI(c_i)$ for the seven corresponding layer (λ-1) clusters with centres c_i, $i = 0,...,6$ such that

$$D_\lambda(s) = \sum_{i=0}^{6} H_1(c_i) \times CI(c_i) \tag{7}$$

Although the framework presented can be used for many image processing algorithms, this paper uses edge detection as its application. To achieve edge detection at a scale corresponding to Layer λ=2, we apply each of the 7 operator values to the cluster integral values as illustrated in Figure 4 and compute a gradient magnitude at each of these cluster integral centres. Once each gradient magnitude value is computed, all of the other pixels in the cluster are assigned the same gradient magnitude, as illustrated by the shading in Figure 5.

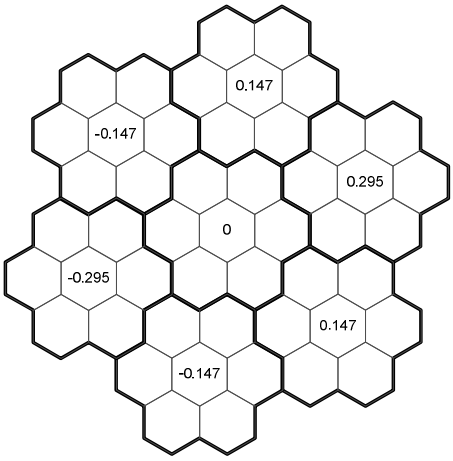

Fig. 4. Convolving 7-point operator with the integral image at Layer 2

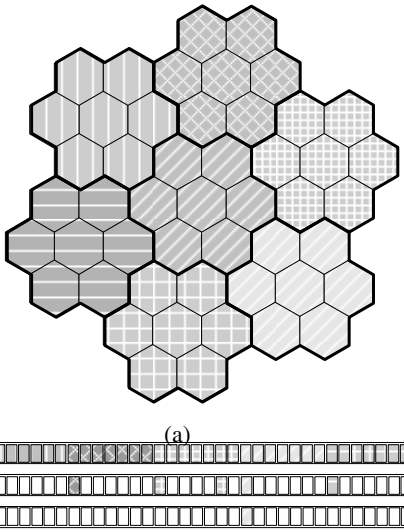

Fig. 5. (a) Representation of the gradient magnitude values within each cluster at Layer 2; (b) Representation of storage required for gradient magnitude values

As a result of the integral cluster approach, as each pixel within a cluster has uniform gradient magnitude, it is necessary only to store each gradient magnitude value computed at the cluster centre, thus requiring only one seventh of the storage space required for the entire image as represented by Figure 5(b). Reconstruction of a complete gradient image from the gradient magnitude values is easily achieved as each location in the storage can be directly mapped to the corresponding spiral address i, in an image vector of size M, using:

$$i + 1 = i + 10^{\lambda-1},\ 0 \bmod 7,\ \text{for } i=0,\ldots,M \tag{8}$$

5 Performance Evaluation

We present results for our proposed hexagonal integral image in comparison with the original approach in [7], and with standard convolution of an operator with a spiral image where the pixel neighbour addresses are stored in a look-up table (this takes 0.4017s to generate, but is significantly faster than standard hexagonal addressing, which requires *mod* 7 arithmetic). In Table 1 and Table 2 we present run times for feature extraction at Layer 1 and Layer 2 respectively. Processing times are computed on a Pentium dual-core workstation using unoptimised C++ code by processing each image 10 times and computing the average time. The time taken to compute the integral image is 0.001751 and this is only required to be computed once to enable multi-scale feature extraction. In Table 1 we present the average run-times for two approaches, of an operator with a spiral image and standard convolution where the pixel neighbour addresses are stored in a look-up table (LUT) and the integral approach at Layer 1. The LUT is an alternative to computing the nodal addresses within a neighbourhood by using hexagonal arithmetic, which is very computationally expensive. The LUT approach effectively pre-computes and stores the indices for all of the 7 pixel neighbourhood clusters. Nodal addresses for neighbourhood clusters at larger scales can be obtained hierarchically by combining the use of the LUT with simple hexagonal arithmetic.

Table 1. Algorithm run-times for 7-point operator ($\lambda = 1$)

Method	Runtime
Standard spiral convolution	3.7621s
Spiral convolution using LUT	0.0711s
Integral Image Approach	0.0018s

Table 2. Algorithm run-times for 49-point operator ($\lambda = 2$)

Method	Runtime
Standard spiral convolution	24.5104s
Spiral convolution using LUT	0.2191s
Integral Image Approach	0.0830s

In Table 2, we present run-times for these two approaches and the integral approach at Layer 2. These run-times demonstrate that the integral image approach is significantly faster than the other existing approaches. In fact, when using the integral images, applying the 7-point operator at any scale λ>=2 will also take only 0.0830 seconds, as each convolution requires only 7 subtractions and 7 multiplications. Hence the technique based on the use of spiral integral images maintains low computational complexity as scale increases. In contrast, 343 multiplications per convolution are required by the other two approaches when λ=3, increasing by a factor of 7 for each increase in scale.

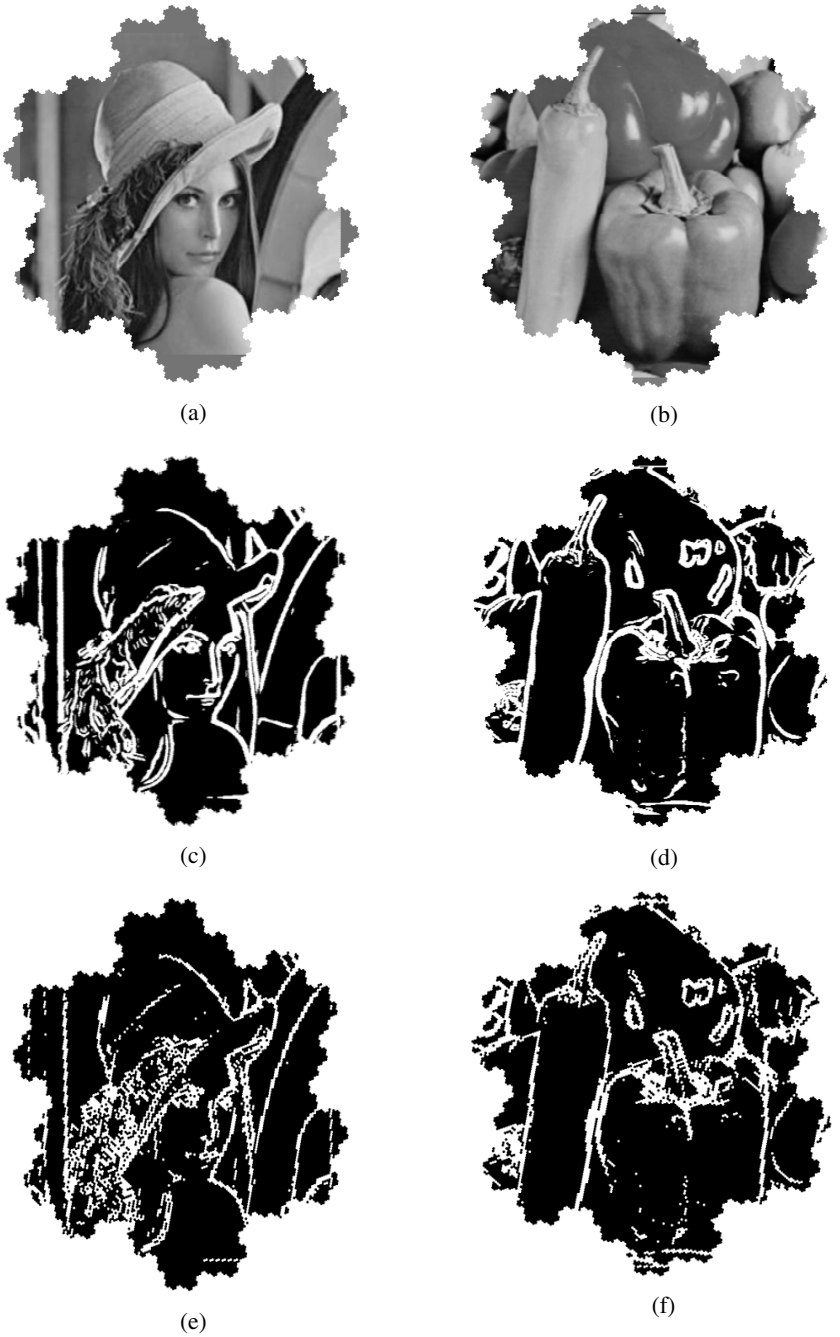

Fig. 6. (a)(b) Original spiral images; (c)(d) 49-point operator applied using Spiral approach; (e)(f) 7-point operator applied using integral eye-tremor approach

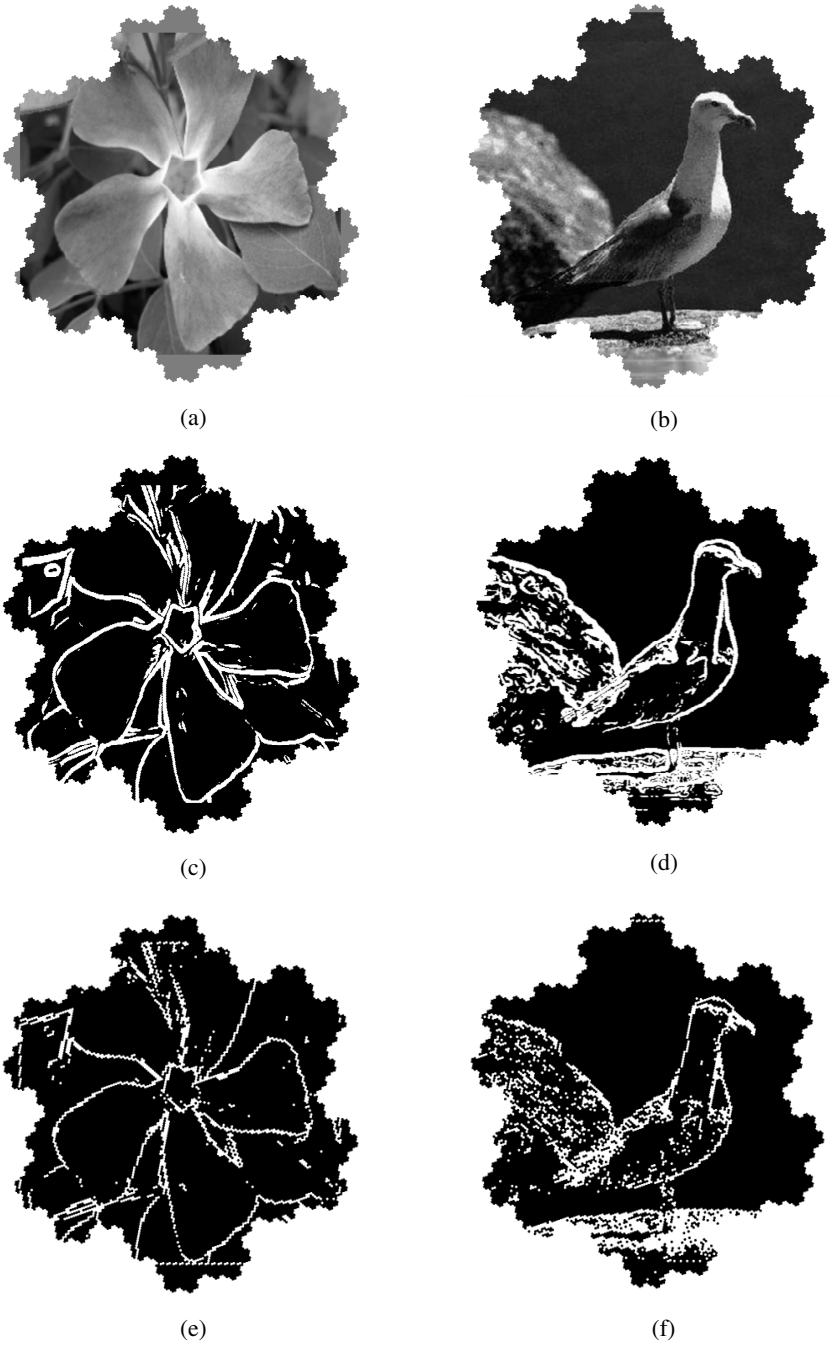

Fig. 7. (a)(b) Original spiral images; (c)(d) 49-point operator applied using Spiral approach; (e)(f) 7-point operator applied using integral eye-tremor approach

Figure 6 and Figure 7 present edge maps obtained by applying a 49-point operator [8] directly to a standard spiral image and the 7-point operator to hexagonal integral images at scale $\lambda=2$. The features detected using the integral images are coarse compared with those detected as each location when applying the 49-point operator, however coarse features detected in real-time may be appropriate of robot navigation and similar applications.

6 Conclusion

We have presented a novel hexagonal integral image that enables fast feature extraction (approximately 12 fps). We have demonstrated that the approach of applying the 7-point operator to the spiral integral image at various scales is significantly faster than applying scaled operators to the original image, as we require only 7 subtractions and 7 multiplications to generate each output value regardless of the scale at which the operator is applied. The visual results are comparable and appropriate for applications requiring fast, coarse feature extraction.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
2. Bradley, D., Roth, G.: Adaptive thresholding using the integral image. J. Graphics, GPU, & Game Tools 12(2), 13–21 (2007)
3. Grabner, M., Grabner, H., Bischof, H.: Fast Approximated SIFT. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 918–927. Springer, Heidelberg (2006)
4. Lindeberg, T.: Feature Detection with Automatic Scale Selection. International Journal of Computer Vision 30, 79–116 (1998)
5. Middleton, L., Sivaswamy, J.: Hexagonal Image Processing; A Practical Approach. Springer (2005)
6. Roka, A.: Edge Detection Model Based on Involuntary Eye Movements of the Eye-Retina System. Acta Polytechnica Hungarica 4(1), 31–46 (2007)
7. Scotney, B.W., Coleman, S.A., Gardiner, B.: Biologically Motivated Feature Extraction Using the Spiral Architecture. In: Proc. IEEE ICIP, pp. 221–224 (2011)
8. Sheridan, P.: Spiral Architecture for Machine Vision. Ph.D. Thesis, University of Technology, Sydney(1996)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, pp. 511–518 (2001)
10. Wang, J.: Coarse-to-Fine Vision Based Localization by Indexing Scale-Invariant Features. IEEE Trans. Systems, Man, and Cybernetics, Part B 36(2), 413–420 (2006)