Hexagonal Structure for Intelligent Vision

Xiangjian He and Wenjing Jia
Computer Vision Research Group
Faculty of Information Technology
University of Technology, Sydney
Australia

Abstract-Using hexagonal grids to represent digital images have been studied for more than 40 years. Increased processing capabilities of graphic devices and recent improvements in CCD technology have made hexagonal sampling attractive for practical applications and brought new interests on this topic. The hexagonal structure is considered to be preferable to the rectangular structure due to its higher sampling efficiency, consistent connectivity and higher angular resolution and is even proved to be superior to square structure in many applications. Since there is no mature hardware for hexagonal-based image capture and display, square to hexagonal image conversion has to be done before hexagonal-based image processing. Although hexagonal image representation and storage has not yet come to a standard, experiments based on existing hexagonal coordinate systems have never ceased. In this paper, we firstly introduced general reasons that hexagonally sampled images are chosen for research. Then, typical hexagonal coordinates and addressing schemes, as well as hexagonal based image processing and applications, are fully reviewed.

I. INTRODUCTION

Since Golay [1], the possibility of using a hexagonal structure to represent digital images and graphics has been studied by many researchers. Hexagonal grid is an alternative pixel tessellation scheme besides the conventional square grid for sampling and representing discretized images. Sampling on a hexagonal lattice is a promising solution which has been proved to have better efficiency and less aliasing [2]. The importance of the hexagonal representation is that it possesses special computational features that are pertinent to the vision process. Its computational power for intelligent vision pushes forward the image processing field. Dozens of reports describing the advantages of using such a grid type are found in the literature. Among these advantages are higher degree of circular symmetry, uniform connectivity, greater angular resolution, and a reduced need of storage and computation in image processing operations.

In spite of its numerous advantages, hexagonal grid has so far not yet been widely used in computer vision and graphics field. The main problem that limits the use of hexagonal image structure is believed due to lack of hardware for capturing and displaying hexagonal-based images. In the past years, there have been various attempts to simulate a hexagonal grid on a regular rectangular grid device. The simulation schemes include those using rectangular pixels, pseudohexagonal pixels, mimic hexagonal pixels and virtual hexagonal pixels. Although none of these simulation schemes can represent the hexagonal structure without depressing the advantages that a real hexagonal structure possesses, the use of these techniques provides us the practical tools for image processing on hexagonal grids and makes it possible to carry on theoretical study of using hexagonal structure in existing computer vision and graphics systems.

The use of hexagonal grid is also fettered by its pixel arrangement. In the hexagonal structure, the pixels are no longer arranged in rows and columns. In order to take the advantages of the special structure of hexagonal grid, several addressing schemes and coordinate systems have been proposed. There exist a 2-axis oblique coordinate system, a 3-axis oblique coordinate system, and a single dimensional addressing scheme.

This paper is organized as follows. In Section II, we list the major reasons to be based on hexagonal structure for intelligent vision system. In Section III, we introduce several typical hexagonal simulation schemes. In Section IV, three addressing schemes on hexagonal structure are demonstrated. Image processing algorithms using hexagonal grid are discussed in Section V.

0-7803-9421-6/05/$20.00 ©2005 IEEE.

II. WHY HEXAGONAL?

Since the introduction of computer graphics, one of the biggest problems that scientists have to face is the fact that the physical screen is a discrete set of points, i.e., a countable set of isolated points, and the real world is in a continuous environment. Moreover, in order to store, process, display and transfer images by digital devices, the image plane must be quantized into spatial elements of finite dimension, generally referred to as *pixels*.

Digitization, which is to convert real images into discrete sets of points, has been therefore one of the earliest subjects of study for computer scientists involved in vision and graphics research. Each point which forms an image on the screen must be properly addressed in order to be indexed. The disposition of the points on the plane, called *digitization scheme*, however, can take different choices. Considering technical implementation, these points must be placed as regularly as possible on the plane and they must be disposed so that the coverage of the plane is as efficient as possible.

A. Three Possible Regular Tessellation Schemes

There exist only three possible regular tessellation schemes to tile a plane without overlapping among the samples and gaps between them, namely the tessellation with hexagons, with squares, and with regular triangles [3, pages 61-64] (see Fig. 1). Any other types of spatial tessellation will result in either unequal distance between neighboring pixels, or introduce gaps or overlaps among samples. A simple explanation is given below. For more detailed proof, please refer to [3, page 61-64].

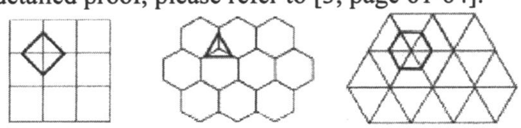


Fig. 1. Three schemes of regular tessellation

We use the symbol $\{p, q\}$ to denote the tessellation of regular p-gons, of which each has q pixels (p-gond) surrounding each vertex. It is easy to see that the $\{p,q\}$ pair[1]s are {4,4} , {6,3} and {3,6} for the three tessellation schemes as illustrated in Fig. 1, where in each case the polygon drawn in bold lines is the *vertex figure*, i.e., the q -gon whose vertices are the midpoints of the q edges connected to a vertex. A tessellation is said to be *regular* if it has regular faces and a regular vertex figure at each vertex.

On the left is the square case {4,4}, which is familiar and usual because it is aligned with the standard Cartesian axes, which helps to make operations simple and intuitive. The far right illustrates the triangular case {3,6} , which yields a denser packing than the square case. This means that more information is contained in the same area of the image. The tessellation in the middle figure, hexagonal case {6,3} , is often used for tiled floors and it can be seen in any beehive. It is believed to be the most efficient tessellation scheme among them.

B. More Efficient Sampling Schemes

No matter which sampling scheme is chosen, an insufficient sampling rate can always introduce unwanted effects in the reconstructed signal, referred as *aliasing*. Peterson and Middleton [5] investigated sampling and reconstructing wave-number-limited multi-dimensional functions and concluded that the most efficient sampling lattice, i.e., which uses a minimum number of sampling points to achieve exact reproduction of a wave-number-limited function[1], is not in general rectangular. Specially, when a two-dimensional isotropic function[2] is considered, the optimum sampling lattice is the 120q rhombic (hexagonal) with spacing of sample points equal to $(\sqrt{3}B)^{-1}$ if the spectrum of a function is bounded by a circle of radius $2\pi B$ in the wave-number plane (see Fig. 2). The sampling efficiency is 90.8%, compared with 78.5% for the largest possible square lattice.

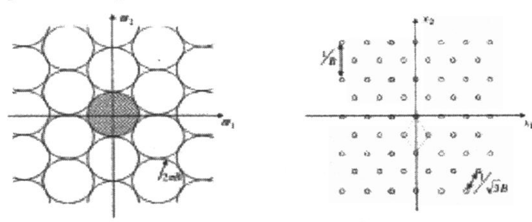


Fig. 2. Optimum sampling lattice for two-dimensional isotropic function

1 We call a function *wave-number-limited* if the Fourier transform of the function lie within a bounded region of spectrum space of the function [5].

2 An *isotropic* function is defined in the broad sense as describing a spectrum which cuts off at the same wave-number magnitude in all directions [5].

A similar conclusion was obtained by Mersereau [2], who developed a hexagonal discrete Fourier transform and hexagonal finite impulse response filters. Mersereau showed that for signals which are band-limited over a circular region in Fourier space, 13.4% fewer sampling points are required with the hexagonal grid to maintain equal high frequency image information with the rectangular grid, thus less storage and less computation time are required. An example is that in image coding application, one may expect that the coding efficiency can be increased by using the hexagonal sampling scheme.

Recently, Vitulli, Armbruster and Santurri [6], after investigating the sampling efficiency of hexagonal sampling, also concluded that using hexagonal sampling, about 13% less number of pixels are needed to obtain the same performance as obtained using square sampling when sampling the same signal.

These conclusions are briefly illustrated below. Fig. 3((a) is a generic hexagonal sampling lattice. *Goodman* [6,7] *showed that the Fourier transform (FT) of a hexagonal lattice is still a hexagonal lattice.* In [7, page 12], however, it is said that *the Fourier transform of a circularly symmetric function is itself circularly symmetric*, where the function f can be said to be circularly symmetric if it can be written as a function of r alone, that is, $g(r, \theta)$ g (r) R T [7, page 11]. In hexagonal lattices, the inverse of the sampling steps that corresponds with the distances between two aligned rows and columns in FT domain are twice the corresponding steps in spatial domain, as shown in Fig. 3(b). Similar to square case, the FT of the hexagonally sampled image is also composed of infinite replicas of the spectrum G (ξ,η) , the FT of the image $g(x, y)$. These replicas are centered in the points of the hexagonal lattice, which is the FT of the hexagonal sampling lattice.

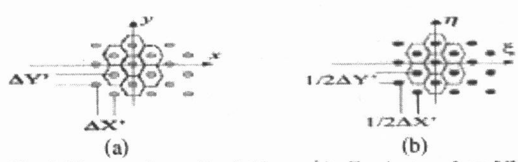


Fig. 3. Hexagonal sampling lattice and its Fourier transform [6]

Vitulli etc. compared the Nyquist constraints, i.e., the minimum sampling densities without aliasing, between rectangular and hexagonal cases. The bigger the minimum sampling density required is, the better the sampling performance will be. The maximum densities to tile spectra in spectrum space are illustrated in Fig. 4, for rectangular grid (left) and for hexagonal grid (right). The pixel density with hexagonal sampling that avoids aliasing is 3 2 and thus is lower than rectangular one.

As a result, using hexagonal grid, wider spectra can be sampled without aliasing with the same number of pixels, or less pixel than using square grid.

B. Smaller Quantization Error

As mentioned earlier, in order to process an image by a digital computer, the continuous image in real world must be quantized into spatial elements of finite dimensions, generally referred as pixels. Due to the limited resolution capabilities of image sensors, this array is usually too small to adequately represent the scene in real world.

Quantization error, thus, is inevitable. In computer vision, quantization error is a very important measurement to investigate the merits of different types of sensory configurations in order to find which spatial sampling would introduce less quantization error into computations. Kamgar-Parsi [8-11] developed formal expressions for estimating quantization error in hexagonal spatial sampling and found that, for a given resolution capability of the sensor, hexagonal spatial sampling yields smaller quantization errors than square sampling. Fig 4. Spectral packaging for best rectangular and hexagonal sampling [6]

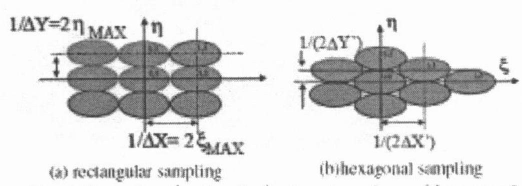


Fig. 4. Spetral packaging for best rectangular and hexagonal sampling [6]

C. Consistent Connectivity Definition

Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To decide if two pixels are connected, it must be determined if they are neighbors and if they satisfy a specified criterion of similarity [12, pages 66-67].

On a square grid, there are two possible ways to define neighbors of a pixel. We can either regard pixels as neighbors when they have a common edge or when they have at least one common corner, so that four and eight neighbors exist (referred as a 4-neighborhood and an 8-neighborhood). Accordingly, on a square grid, object connectivity can be defined as 4-way to any of the four nearest neighbours, or 8-way if connectivity to diagonal neighbours is permitted. This is illustrated in Fig. 5.

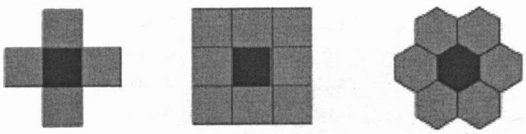


Fig. 5. Neighbourhood relationship on the square grids and hexagonal grids

Correspondingly, background connectivity must be 8-way if object connectivity is four-way or 4-way if object connectivity is 8-way [13]. Verification of this statement is presented below.

Consider the pattern shown in Fig. 6(a). Assuming 4-way connectivity for both the object and the background, the number of vertices V in the pattern is 16, the number of edges E is 16, and the number of faces F is 4.

Application of the Euler formula $V= E+ F$ to the pattern should give its *genus*. Thus, by the above formula the genus is16-16+4 = 4. However, the pattern has four objects and the background has two, so that the genus (the number of object components minus the number of background components +1) is 4 – 2 +1= 3. A similar disagreement in the value of the genus arises when 8-way connectivity is assumed. For then the number of vertices V in Fig. 6(a) is 12, the number of edges E is 16, and the number of faces F is 4. Thus, by Euler's formula, the genus is 0, whereas in fact it should be 1.

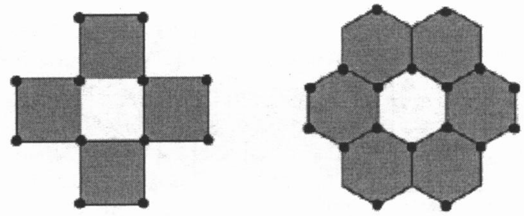

(a) (b)

Fig. 6. Square grid and hexagonal grid

However, if 4-way connectivity is assumed for the object and 8-way connectivity for the background then, according to Euler's formula, the genus is 4. Since the number of background components is now 1 (not 2), the value of the genus obtained by counting the number of components is also 4. Similarly, when we assume the pattern to be 8-neighbor connected and the background to be 4-neighbor connected, both methods of calculating the value of the genus, which is 0, agree.

The hexagonal grid, however, offers no connectivity choice. We can only define a 6-neighborhood. Neighboring pixels have always one common edge and two common corners (see Fig. 6(b)). The absence of such choice in hexagonal grid results in easier and more efficient algorithms, such as thinning algorithm [13][14][15], since fewer connectivity situations have to be accounted for. Accordingly, connectivity in hexagonal objects is consistent as it is six-way to either of the nearest neighbours for both the object and the background image components [16, 17].

Assuming 6-neighbor connectivity, the number of vertices V in Fig. 6(b) is 24, the number of edges E is 30 and the number of faces F is 6. Using the Euler formula, the genus is equal to 0. Since the number of components of the pattern is 1 and the number of components of the background is 2, the genus has the value1-2+1= 0. Thus, both values of the genus agree.

D. Equidistance

With the introduction of neighbourhood relation, distance function can be easily measured. In square grid we have two types of distances, where the distance between adjacent pixels in the diagonal direction is √2 times of that in the horizontal (or vertical) direction (see Fig. 7(a)).

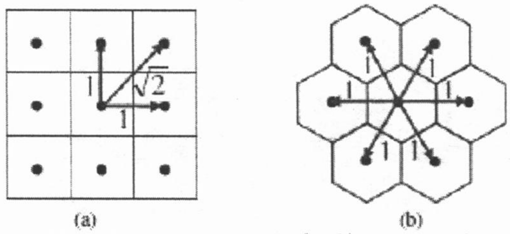


Fig. 7. Distance in a square grid and hexagonal grid

While in hexagonal case, each hexagonal pixel has and only has six neighboring pixels and each pixel is equidistantly adjacent to their six neighbors along the six sides of the pixels. The centroid of the middle pixel is at the same distance from the centroids of the six adjacent pixels (see Fig. 7(b)).

E. Greater Angular Resolution

Image processing on a hexagonal lattice is advantageous is also believed due to its greater angular resolution to represent curved objects. It has been noted that hexagons offer greater angular resolution as the nearest neighbors of the same type are separated by 60q instead of 90q [1]. An example showing a familiar curved figure and a representation on square and hexagonal lattices is shown in Fig. 8.

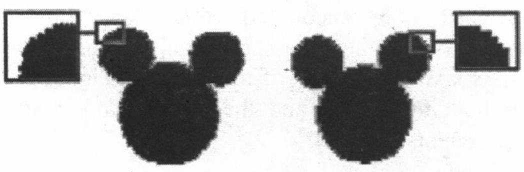


Fig. 8. Curved figure represented in hexagonal grid and hexagonal grid

Notice that the hexagonal case, on the left of Fig. 8, appears to have smoother curves than the square case. There are several reasons for this. The first is due to the consistent connectivity in the hexagonal lattice. This means that all neighbours are uniform distances away from each other and leads to the smoother curvature. Another reason is what is known as the oblique effect in human vision (see web link http://www.ecs.soton.ac.uk/~ljm/hip.php). This means that we have a visual preference for lines at oblique angles. This also helps to make the hexagonal curves look smoother.

As a matter of fact, the theory developed and the simulation done on a physical screen in [4] showed that hexagonal grids represent a reasonable alternative to conventional square grid display techniques not only for circle drawing, which was somehow predictable, but also for straight lines.

On the hexagonal grid, digitizations display a better connectivity and are perceived as being approximated by small polylines, whereas on the square grid, digitizations are still perceived as being approximated by pixels. Such a perception of single pixels disturbs the impression of continuity of the discretized line. This is due to the fact that in the square grid neighbors of a pixel are not placed all at the same distance. Moreover, two diagonal neighbors in the square grid have only one point in common, whereas two horizontal or vertical neighbors of the square grid, and all the neighbors of a pixel in the hexagonal grid, have one segment in common with their neighbor. This fact produces thickness variations in square digitizations, leading to greater edge busyness and to a thinner average width in a line digitization.

F. Higher Symmetry

Serra [19] has developed many of morphological operators that were currently used for image processing. He prefers the hexagonal grid to the rectangular because of the connectivity definition and the higher symmetry, which lead to simpler processing algorithms. It can be seen in Fig. 6 that the cluster of hexagonal pixels possesses the same symmetry about the three different lines connecting pairs of two pixels and the central pixel. This symmetry degree is one higher than that of square grid. This symmetric feature makes image processing more accurate. For example, when an image on a hexagonal grid is rotated, more image information will be retained compared to the same rotation is performed on square grid.

G. Other Features of Hexagonal Grid

The research done in the biological domain of animal vision clearly demonstrates that in animal vision systems the arrangement of rods and cones in the fovea more nearly approximate a hexagonal tessellation than a rectangular one. Specifically the research by Hubel [20] shows that the fovea can nearly be described by a regular hexagonal tessellation. Another compelling reason to investigate other tessellations of the plane is the well known paradox concerning the definition of the nearest neighbor network such that edges are continuous and that the inside of an object not be connected to the outside of the same object. [21]

III. HEXAGONAL IMAGE REPRESENTATION

In spite of the many advantages of hexagonal structure, the hexagonal based image processing has not been used widely in intelligent vision area. The main reason is that currently there is no hexagonal-based device available to capture and display digital images on hexagonal grids. So how to simulate hexagonally sampled images on common square display equipments has once become a serious problem that affects the advanced research on hexagonal architecture in the field of computer vision and graphics.

Fortunately, there have been several ways to simulate a hexagonal grid on a regular rectangular grid. We list three most common simulations as follows. The use of these techniques allows us to take the advantages of hexagonal grids for computer vision and computer graphics.

A. Mimic Hexagonal Pixels Using Square Pixels

Horn [22] has described how a practical hexagonal data may be captured by delaying sampling by half a pixel width on alternate TV scan lines in horizontal direction (see Fig. 9). In his scheme, the pixel shape is square. In other words, the sampling intervals in horizontal and vertical directions are identical. This scheme simplifies the hardware design by setting identical sampling intervals in both horizontal direction and vertical direction. However, the equidistance property of hexagonal pixels is not preserved. A shown in Fig.9, if we denote the distance between any two neighbors in horizontal and vertical direction as 1 unit, the distance between any two neighboring pixels in diagonal direction will be √3/2.

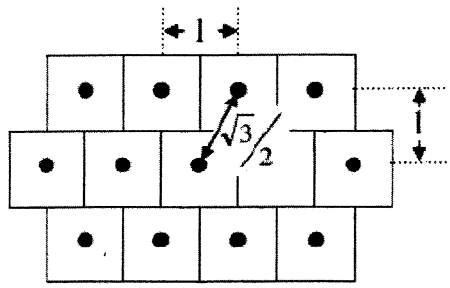


Fig. 9. Using half-pixel shifted square pixels represent hexagonal structure

Later on, Staunton [23] described a hexagonal data structure with a rectangular shape, where the sic neighboring pixels of a pixel all lie on a circle with the centre of the circle being at the sampling point of the central pixel, as illustrated in Fig.10. The major advantages with this structure are that, all sampling points are equidistant from their nearest neighbors, the angle subtended by two nearest neighboring points is 60°, and |the horizontal sampling distance is 2/√3. The pixel size is one by 2/√3 and thus for systems employing an equal number of pixels horizontally and vertically, the image aspect ratio would be 2/√3 :1 .

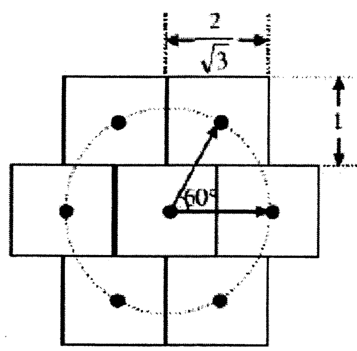


Fig. 10. Rectangular pixels on a hexagonal sampling grid

B. Pseudo Hexagonal Pixel

Wuthrich [4] proposed a pseudo hexagonal pel (see Fig. 11) in order to evaluate the visual effect of hexagonal pixel and square pixel. A comparative simulation of two screens based on the square and the hexagonal lattices has been made. A hexagonal pixel, called a hyperpel, is simulated using a set of many square pels and the simulated square grid had to be adapted in order to make its density comparable with the hexagonal grid. This results in a great loss in the screen resolution and to an inexact simulation of the square grid, reducing it to a rectangular grid. In order to approximate the square and the hexagonal grids, two ideal lattices has been selected, that is $\Lambda_6 = \Lambda((\sqrt{3}/2, 1/2), (0,1))$ and $\overline{\Lambda} = \Lambda((\sqrt{3}/2, 0), (0,1))$ as an approximation of Λ_4. $\overline{\Lambda}$ has been chosen such that the points of the two lattices have the same density, i.e., the same number of points per unit of surface. As there is no way to display exactly the point $(\sqrt{3}/2, 1/2)$ on the square grid, in the practical simulator the lattices hat have been actually drawn on the simulator are thus

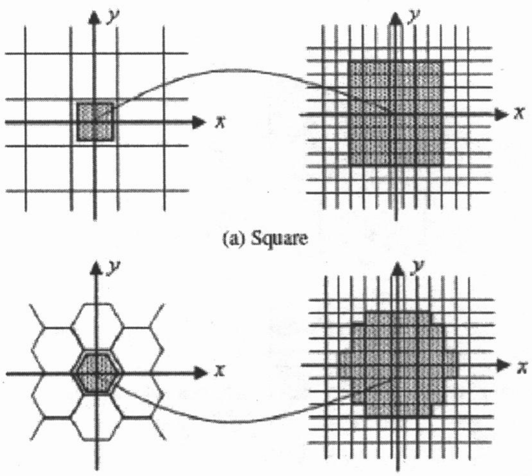
$$\overline{\Lambda}_4 = \Lambda((7/8, 0), (0,1))$$

and

$$\overline{\Lambda}_6 = \Lambda((7/8, 1/2), (0,1))$$

respectively for the square and the hexagonal lattices. The resulting hyperpels, which are illustrated in Fig. 11 were displayed at a resolution of 60 u 60 pels.

This idea has been adopted by Yabushita in [18] who designed a similar pseudo hexagonal picture (hexelement), which is also composed of small square pixels and which aspect ratio is 12:14.



(a) Square

(b) Hexagon

Fig. 11. Simulated hyper pixel

Simulated hyper pixel Middleton and Sivaswamy [24, 25] proposed a framework for practical hexagonal-image processing, where a process known as *image re-sampling* is employed to generate a hexagonally sampled image from normal square image.

C. Mimic Hexagonal Structure

He [26] proposed a mimic hexagonal structure, called *mimic Spiral Architecture*, where one hexagonal pixel consists of four traditional square pixels and its grey level value is the average of the involved four pixels (see Fig. 12). This mimic scheme preserves the important property of hexagonal architecture that each pixel has exactly six surrounding neighbours. However, because the grey-level value of the mimic hexagonal pixel is taken from the average of the four corresponding square pixels, this mimic scheme introduces loss of resolution. In addition, we know that according to hexagonal structure theory the distance between each of the six surrounding pixels and the central pixel is the same. However, this property is lost in the mimic Spiral Architecture.

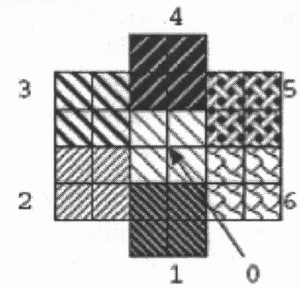


Fig. 12. A cluster of 7 mimic hexagons

D. Virtual Hexagonal Structure

Later, Wu [27] constructed a virtual hexagonal structure which is an important milestone for the theoretical research and the practical application exploration of this architecture. Using virtual Spiral Architecture, images on rectangular structure (or called square grid as indicated in Fig. 13) can be smoothly converted to Spiral Architecture. Such virtual Spiral Architecture only exists during the procedure of image processing. It builds up a virtual hexagonal grid system on memory space on computer. Then, processing algorithms can be implemented on such virtual spiral space. Finally, resulted data can be mapped back to rectangular architecture for display (see Fig. 13). Unlike the previously proposed mimicking methods, this mimicking operation nearly does not introduce distortion or reduce image resolution, which is the most remarkable advantage over other mimicking methods, while keeping the isotropic property of the hexagonal architecture.

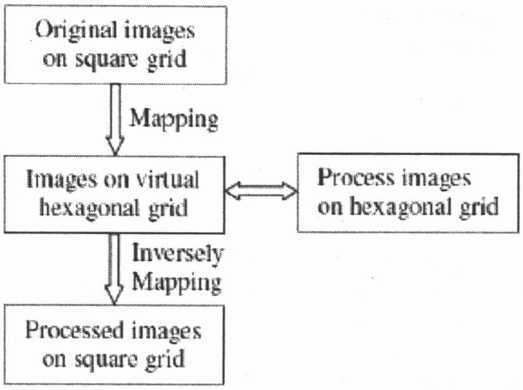


Fig. 13. Image processing on virtual Spiral Architecture

IV. HEXAGONAL STRUCTURE ADDRESSING

Obviously, no matter which kind of simulating scheme is chosen, there exists another big problem that the hexagonal pixels cannot be labeled in normal column-row order as in rectangular grid. In order to properly address and store hexagonal images data, different coordinate systems have been proposed. In this section, typical coordinate systems are reviewed.

A. 2-Axes Oblique Coordinate Addressing Scheme

Using two oblique axes (see Fig. 14) to address hexagonal structure is firstly suggested by Luczak and Rosenfeld [28], also referred as *skewed coordinate system* in [4], and *h2 system* in [29], where two basis vectors are not orthogonal. With such an oblique coordinate system, each hexagonal pixel can be addressed by an ordered pair of unit vectors, **u** and **v**, as illustrated in Fig. 14, which indicate a horizontal deflection and an upright deflection respectively. The system has been shown to have the following properties:

1. *Complete:* Be sufficient to represent any point in a 2-dimensional space;
2. *Unique U:* Any ordered pair corresponds to exactly one point;
3. *Convertible t:* It can be easily converted to and from Cartesian coordinate; and
4. *Efficient:* It is a convenient and efficient representation.

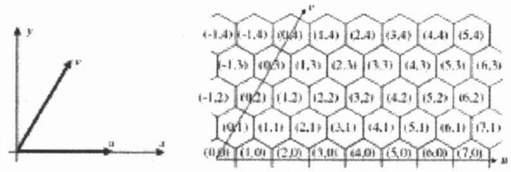


Fig. 14. 2-axis oblique coordinate system for hexagonal structure. Unit vectors *u* and *v* describe this coordinate system

B. Three-Coordinate Symmetrical Coordinate Frame

In [30,31], Her developed a *symmetrical hexagonal coordinate frame*, denoted as *$R3$, for hexagonal grid, which uses three coordinates x, y, z, instead of two, to represent each pixel on the grid plane, as shown in Fig. 15. The three coordinates at any pixel has a relationship among them:

$$x + y + z = 0.$$

Here the distance between two neighboring grid points is defined as one unit.

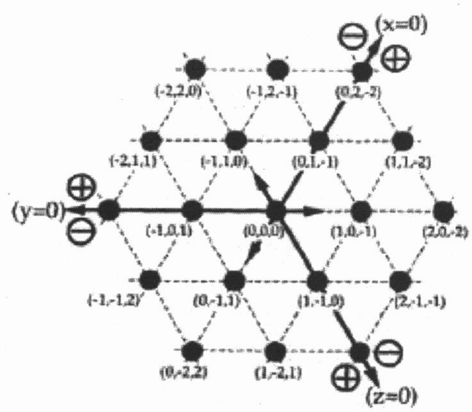


Fig. 15. Symmetrical hexagonal frame *$R3$

The major advantage of this coordinate system is that there is a one-to-one mapping between *R^3 and the 3-dimensional Cartesian frame R^3, as illustrated in Fig.16, where, x, y and z are the three orthogonal axes of R^3. Due to this reason, many geometrical properties of R^3 can be readily transferable to *R^3. Moreover, since the x and y coordinates of a point of this symmetrical hexagonal coordinate frame*R^3 are actually the two coordinates used in the oblique coordinate frame (see Fig. 14), theories and equations previously developed for the oblique coordinate frame can directly be used in*R^3. Moreover, in [32], the use of this symmetrical hexagonal coordinate frame is demonstrated to derive various affine transformations.

Due to the physical relationships between the symmetrical hexagonal coordinate frame and the 3-dimensional Cartesian frame R^3, geometric transformations on the hexagonal grid are conveniently simplified and the beautiful symmetry property of the hexagonal grid is successfully preserved.

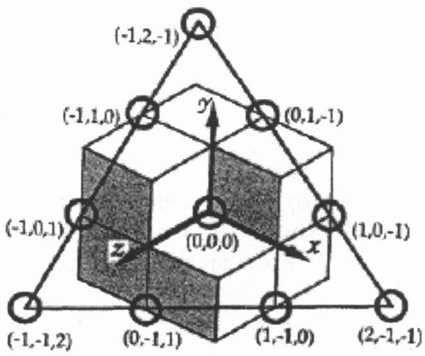


Fig. 16. Relation between frames $^*R3$ and $R3$

This three-axis coordinate system is used in [33] for mathematically handling the hexagonal structure, for example, numerically calculating the distance of two objects. This three-axis coordinate system reflects the geometrical symmetry of the grid.

C. Single Indexing System

Sheridan [34] proposed a one-dimensional addressing system, as well as two operations based on this addressing system, for hexagonal structure. This system is called Spiral Architecture (see Fig. 17). Spiral Architecture (SA) is inspired from anatomical consideration of the primate's vision system.

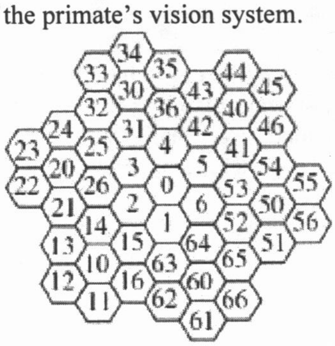


Fig. 17. Spiral addressing

Sheridan [34] presented a one dimensional indexing scheme, called *Spiral Addressing*, to address each hexagon on the image. This address grows from the centre of image in powers of seven along a spiral like curve. This addressing scheme combined with two later proposed mathematic operations, spiral addition and spiral multiplication is the basic of *Spiral Architecture* [26,34]. The spiral addition and spiral multiplication correspond to image translation and image rotation respectively.

Middleton and Sivaswamy [24,25] also proposed a similar single-index system for pixel addressing by modifying the Generalized Balanced Ternary system, as shown in Fig. 18.

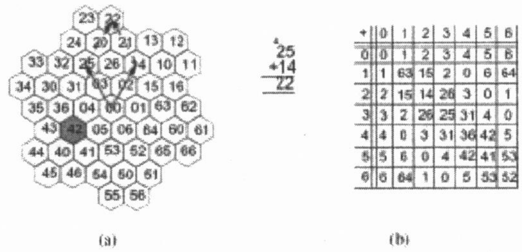


Fig. 18 (a) The hexagonal image structure with indices; (b) Balanced ternary addition

Neighbourhood operations are often used in image processing. Finding the neighbour in a hexagonal image makes use of the spiral addition operation, of which details can be found in [34]. In a seven-pixel cluster, the neighbourhood relation can be determined by spiral addition as follows.

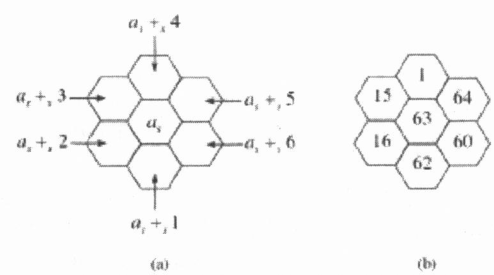


Fig. 19. Neighbourhood relationship within Spiral Architecture. (a) Neighbourhood Relationship; (b) An example of neighbourhood

Let the spiral address of the central pixel, as shown in Fig. 19(a), be denoted by a_s. Then the spiral address of its neighbour pixel can be described by spiral addition denoted by $+_s$ with a certain number of displacements, as shown in Fig. 19 (a). An example is given in Fig. 19(b).

For the whole image, following the spiral rotation direction, as shown in Fig. 20, one can find out the spiral address of any hexagonal pixel with centre on a certain hexagonal pixel whose spiral address is known.

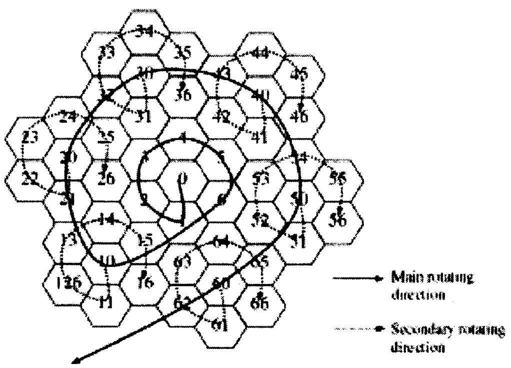


Fig. 20. Spiral rotating direction

The Spiral Architecture has some distinguishing features compared to the square image processing. First, the one dimensional addressing scheme leads to an efficient storage and the placement of the origin at the centre of the image simplifies geometric transformations of a given image. Finally, the hexagonally sampled image allows non-traditional neighbourhoods with consistent boundary connectivity, which is useful for many computer vision applications.

V. HEXAGONAL IMAGE PROCESSING

Although hexagonal image representation and storage has never yet come to any standard, theoretical studies on hexagonal image processing have never ceased.

A. Hexagonal Image Transformation

For the purpose of efficient and fast processing and analysing, digital images that are originally defined in spatial domain usually need to be transformed into another domain with certain transformation and take use of some unique characters of the transformed domain to process the transformed image in the domain. Image transformation is the basis of many image processing and analysing techniques. After transformation, processing in spatial domain can be converted into the corresponding processing in transformed domain, which has many advantages. Among them, the most important ones are the computation will be greatly reduced and various image filtering techniques can be applied for image processing. For example, convolution operation becomes more computationally efficient when computed in frequency domain. Further examples include transform coding for image compression purpose and filter design.

Among various image transformations, one of the most widely used is Fourier transformation, which transforms images from spatial domain to spectrum domain. Standard fast Fourier transform (FFT) algorithms, however, are not applicable to non-rectangularly sampled data.

Mersereau [35] developed a two-dimensional fast Fourier transform (2-D FFT) for use with hexagonally sampled data. Nel [21] followed his derivation and corrected a number of algebraic errors in his derivation and derived the 2-D Walsh transformation. They were all derived in non-orthogonal axes. Later, Ehrhardt [36] claimed that Mersereauts 2-D FFT algorithm would require an additional interpolation step, which might introduce artifacts. He presented a separable fast discrete Fourier transform algorithm where the data space is sampled with hexagonal grids and transform space is sampled with rectangular grids. In [37], Middleton derived a Fast Fourier Transform (FFT) for the hexagonal lattice based upon the Cooley-Tukey approach [38] and the radix-7 decimation in space algorithm.

In [39], a hexagonal discrete cosine transform which can be used in the applications of image coding is described and showed that the proposed HDCT is more efficient in energy compaction than the HDFT.

B. Edge Detection on Hexagonal Structure

When a scene is observed by a human, the human visual system first segments the scene. Edge detection is an important approach for image segmentation in computer vision systems. This approach measures the rate of change and decides the existence of an edge at each point. The basic assumption used in most edge detection algorithms is that the edges are characterised by large (step) changes in intensity (or color in color images case). Hence, at the location of an edge, the first derivative of the intensity function should be a maximum or the second derivative should have a zero-crossing.

Middleton [24] investigated the performance of using a hexagonally sampled structure for implementing classical edge detectors, including Prewitt, Laplacian of Gaussian (LoG) and the Canny edge detector. Images that contain curves and straight lines along with a variation in contrast are used for test. Equivalent edge detection masks have been designed for hexagonal images, where the horizontal mask for the hexagonal case is equivalent to the square case, the vertical direction gradient mask is approximated by a combination of two masks oriented at 60° and 120° to the horizontal. Fig. 21 and Fig. 22 give different masks used in the Prewitt edge detector implementation.

Cho [40] applied the edge relaxation to the hexagonal grids. His experiments showed that hexagonal edge relaxation can detect better edges than conventional edge relaxation. This comes from the advantages of hexagonal sampling and unambiguous classification of edge types.

Furthermore, if a closed boundary is reached, then it is unchanged permanently and the open boundary is weakened as the iteration proceeds from the tail of the boundary. Therefore the overall results are reliable in finding the edges in the given edges.

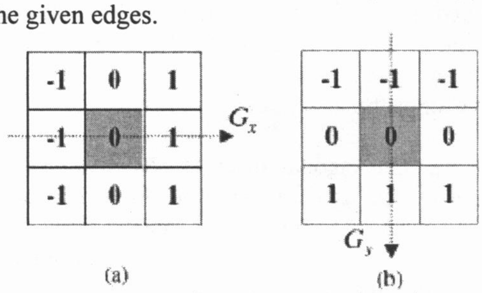


Fig. 21. Prewitt masks used to compute the gradient at the central shadowed point in (a) x -direction, and (b) y -direction on square architecture

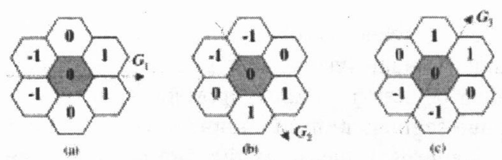


Fig. 22. Prewitt masks used to compute the gradient at the central shadowed point in 3 directions on hexagonal architecture

Several papers on edge detection based on Spiral Architecture have been proposed since 1996. In [41], an overview on edge detection within Spiral Architecture was given. In [42], edge detection using edge focusing technique was proposed. The second edge detection method proposed by Zhou et al [43] applied a bilateral filter which combines a domain filter with a range filter to suppress image noise for edge detection. Another method for edge detection on Spiral Architecture, as shown in [44], was based on triple-diagonal gradient. The gradient of grey-level function was defined as a combination of three vectors in three diagonal directions of hexagonal image structure. This method is a more accurate detection mechanism where the gradient is implemented in a more accurate way in the discrete image space.

Results of edge detection on hexagonal images show that an edge map with better fidelity for curved objects is obtained than with square images. In the case of straight edged objects the edge-maps are of similar quality. This is due mainly to the connectivity of the individual hexagonal pixels generating more consistent contours. Furthermore, using Spiral Architecture for edge detection has computational advantages in order to achieve similar detection results. In particular, convolution operations which are routinely used in edge detection can be implemented with great efficiency. The two together make a strong case for hexagonal based edge detection and seem to reinforce the point that hexagonal image processing can be a viable alternative to conventional square image processing.

C. Hexagonal Thinning

Thinning is the process which is used to reduce the amount of data of an object to obtain its skeleton, which contains single pixel wide lines and can represent the shape of the object. Thinning has been applied to a great variety of patterns in the field of machine recognition [45]. Wide range of applications show the usefulness of reducing patterns to thin-line representations, which can be attributed to the need to process a reduced amount of data, as well as to the fact that shape analysis can be more easily made on line-like patterns. The thin-line representation of certain elongated patterns, for example, characters, would be closer to the human conception of these patters; therefore, they permit a simpler structural analysis and more intuitive design of recognition algorithms. A skeleton should have the following properties [14, 15]:

1. It contains a number of single pixel lines;
2. Each element is connected to at least one other with no gaps in its structure;
3. Skeletal legs are preserved;
4. It is accurately positioned;
5. Noise induced perimeter pixels are ignored and limbs are not formed towards them.

Thinning algorithms for use with rectangular, hexagonal, and triangular arrays has been investigated by Deutsch [13]. He used the same approach to develop each algorithm, where unnecessary pixels were iteratively deleted until no more pixels can be removed. Experimental results on handwritten character recognition showed that the algorithm operating in the hexagonal grid was the most computationally efficient. The resulting thinned images which are obtained using the triangular array contain the least number of points per image, since on this array the neighbors span the largest distance. The ratio of the maximum distances of any neighbor on the rectangular, hexagonal, and triangular arrays is $1: \sqrt{2} : 3 / \sqrt{2}$. However, the increased size of the basic window renders the processing on a triangular array, and thus the resulting image, very sensitive to edge irregularities and, more important, to noise. From this point of view, the hexagonal array is preferential, since all its neighbors, theoretically at least, are equidistant. Moreover, if the thinned image is to be chain encoded the number of direction vectors, in the triangular array is 12, which means that the maximum number of bits required to represent a single direction vector is four; this compares with the three bits required for the other two arrays. So for storage or transmission of a complete resulting line drawing, the triangular array will only be useful if the quarters the number of points in the image on the hexagonal array.

Staunton [14] presented an analysis of the thinning operation from hexagonally sampled images and compared the algorithm experimentally to a similar parallel algorithm designed for a rectangular grid. He defined a set of structuring masks in order to decide whether a pixel could be deleted from the object border. The hexagonal thinning algorithm requires only six masks containing seven elements each, while the rectangular algorithm requires eight masks containing nine elements each. This greatly reduced the processing time for 55% of that required to process the rectangular scheme skeleton. Experimental results also showed that the hexagonal skeleton exhibited more accurate corner representation, noise immunity.

D. Hexagonal Interpolation: Hex-splines

Van De Ville etc. [46,47] constructed a new family of hex-splines which are specifically designed for hexagonal lattices and make use of these splines to derive the leastsquare reconstruction function. Hex-splines are a new type of bivariate splines that are especially designed for hexagonal lattices. Inspired by the indicator function of the Voronoi cell, they are able to preserve the isotropy of the hexagonal lattice (as opposed to their B-spline counterparts). They can be constructed for any order and are piecewise-polynomial (on a triangular mesh). Analytical formula have been worked out in both spatial and Fourier domains. For orthogonal lattices, the hexsplines revert to the classical tensor-product B-splines. While the standard approach to represent two-dimensional data uses orthogonal lattices, hexagonal lattices provide several advantages, including a higher degree of symmetry and a better packing density. They discussed how to advantageously apply them for image processing. We show examples of interpolation and least-squares resampling.

Yabushita [18] investigated the performance of image reconstruction on hexagonal grid. Conventional image reconstruction methods are implemented on square structure. However, on a square grid, the distance between adjacent pixels is different in the horizontal (or vertical) direction from that in the diagonal direction. This difference introduces inconsistency when neighboring pixel values are interpolated with a spherically symmetric weighting function which weight depends on the distance between a given position and the central pixel. Yabushita compared the accuracy of the reconstructed images and compared the results with those obtained on square grid. His experimental results on disc-shaped images showed a better reconstruction quality on hexagonal grid than that on rectangular grid.

REFERENCES

[1] Golay, M., Hexagonal parallel pattern transformation. IEEE Transactions on omputers, 1969. **18**(8): p. 733-740.
[2] Mersereau, R.M., The processing of Hexagonally Sampled Two-Dimensional Signals. Proceedings of the IEEE, 1979. **67**: p. 930-949.
[3] Coxeter, H.S.M., Introduction to geometry. 1969, New York: Wiley.
[4] Wuthrich, C.A. and P. Stucki, An algorithmic comparison between square- and hexagonal-based grids. CVGIP: Graphical Models and Image Processing, 1991. **53**(4): p. 324-339.
[5] Petersen, D.P. and D. Middleton, Sampling and Reconstruction of Wave-Number-Limited Functions in NDimensional Euclidean Spaces. Information and Control, 1962. **5**: p. 279-323.
[6] Vitulli, R., et al., Aliasing effects mitigation by optimized sampling grids and impact on image acquisition chains. Geoscience and Remote Sensing Symposium, 2002.

IGARSS '02. 2002 IEEE International, 2002. **2**: p. 979-981 vol.2.

[7] Goodman, J.W., Introduction to fourier optics. 1968: San Fran. : Mcgraw-hill.

[8] Kamgar-Parsi, B.R. and B.Z. Kamgar-Parsi, Evaluation of quantization error in computer vision. Computer Vision and Pattern Recognition,1988. Proceedings CVP '88., Computer Society Conference on, 1988: p. 52-60.

[9] Kamgar-Parsi, B. and W.A. Sander, III, Quantization error in spatial sampling: comparison between square and hexagonal pixels. Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on, 1989: p. 604-611.

[10] Kamgar-Parsi, B., Evaluation of quantization error in computer vision. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1989. **11**(9): p. 929-940.

[10] Kamgar-Parsi, B., Quantization error in hexagonal sensory configurations. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1992. **14**(6): p. 665-671.

[12] Gonzalez, R.C. and R.E. Woods, Digital image processing. 2002, New Jersey: Prentice Hall.

[13] Deutsch, E.S., Thinning algorithms on rectangular, hexagonal, and triangular arrays. Communications of the ACM, 1972. **15**(9): p. 827-837.

[14] Staunton, R.C., An analysis of hexagonal thinning algorithms and skeletal shape representation. Pattern Recognition, 1996. **29**(7): p. 1131-1146.

[15] Staunton, R.C., A one pass parallel hexagonal thinning algorithm. Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465), 1999. **2**: p.841-845 vol.2.

[16] Mylopoulos, J.P. and T. Pavlidis, On the topological properties of quantized spaces, I. the notion of dimension. Journal of the ACM (JACM), 1971. **18**(2): p. 239-246.

[17] Mylopoulos, J.P. and T. Pavlidis, On the topological properties of quantized spaces, II. connectivity and order of connectivity. Journal of the ACM (JACM), 1971. **18**(2): p. 247-254.

[18] Yabushita, A. and K. Ogawa, Image reconstruction with a hexagonal grid. Nuclear Science Symposium Conference Record, 2002 IEEE, 2002. **3**: p. 1500-1503 vol.3.

[19] Serra, J., Introduction to Mathematical Morphology. Computer Vision, Graphics, and Image Processing, 1986. **35**: p. 283-305.

[20] Hubel, D.H., Eye, Brain and Vision. 1988, New York: Scientific Amprirxn Books.

[21] Nel, A.L., Hexagonal image processing. Communications
and Signal Processing, 1989. COMSIG 1989. Proceedings., Southern African Conference on, 1989: p. 109-113.

[22] Horn, B.K.P., Robot Vision. 1986: MIT Press, Cambridge, MA & McGraw-Hill, New York, NY.

[23] Staunton, R., The design of hexagonal sampling structures for image digitization and their use with local operators. Image and Vision Computing, 1989. 7(3): p. 162-166.

[24] Middleton, L. and J. Sivaswamy, Edge detection in a hexagonal-image processing framework. Image and Vision Computing, 2001. **19**(14): p. 1071-1081.

[25] Middleton, L. and J. Sivaswamy, Framework for practical hexagonal-image processing. Journal of Electronic Imaging, 2002. **11**(1): p. 104-114.

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

[27] Wu, Q., X. He, and T. Hintz, Virtual Spiral Architecture. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2004. **1**: p. 399-405.

[28] Luczak, E. and A. Rosenfeld, Distance on a Hexagonal Grid. IEEE Transactions on Computers, 1976. **25**(5): p. 532-533.

[29] Snyder, W.E., H. Qi, and W.A. Sander, Coordinate system for hexagonal pixels. The International Society for Optical Engineering, Proceedings of SPIE, 1999. **3661**: p. 716.

[30] Her, I., A Symmetrical Coordinate Frame on the Hexagonal Grid for Computer Graphics and Vision. ASME Journal of Mechanical Design, 1993. **115**(3): p. 447-449.

[31] Her, I. and C.T. Yuan, Resampling on a Pseudohexagonal Grid. CVGIP: Graphical Models and Image Processing, 1994. **56**(4): p. 336-347.

[32] Her, I., Geometric transformations on the hexagonal grid. Image Processing, IEEE Transactions on, 1995. **4**(9): p. 1213-1222.

[33] Nagy, B., Finding shortest path with neighbourhood sequences in triangular grids. Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on, 2001: p. 55-60.

[34] Sheridan, P., Spiral Architecture for machine vision. 1996, University of Technology, Sydney.

[35] Mersereau, R.M., A Two-dimensional FFT for Hexagonally Sampled Data(in Digital Signal Processing - ed. V. Cappellini and A.G. Constantinides) Academic Press. 1980: p. 93 - 101.

[36] Ehrhardt, J.C., Hexagonal fast Fourier transform with rectangular output. Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], 1993. **41**(3): p. 1469-1472.

[37] Middleton, L. and J. Sivaswamy, The FFT in a Hexagonalimage Processing Framework. Proceedings of Image and Vision Computing New Zealand, 2001: p. 231-236.

[38] Mersereau, R. and T. Speake, A unified treatment of Cooley-Tukey algorithms for the evaluation of the multidimensional DFT. Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 1981. **29**(5): p. 1011-1018.

[39] Wu, H.-S., Hexagonal discrete cosine transform for image coding. Electronics Letters, 1991. **27**(9): p. 781-783.

[40] Cho, T.I. and K.H. Park, Hexagonal edge relaxation. Electronics Letters, 1992. **28**(4): p. 357-358.

[41] He, X., T. Hintz, and Q. Wu, Edge Detection on Spiral Architecture: an Overview. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2004. **1**: p. 406-412.

[42] He, X. and T. Hintz, Refining edge detection within spiral architecture. Computer Science Conference, 2000. ACSC 2000. 23rd Australasian, 2000: p. 113-119.

[43] Zhou, J., Q. Hu, and X. He, Edge Detection with Bilateral Filtering in Spiral Space. Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004), 2004. **1**: p. 222-225.

[44] Wu, Q., X. He, and T. Hintz, A Triple-Diagonal Gradient- Based Edge Detection. Proceedings of the 6th IASTED International Conference on Computers, Graphics, and Imaging (CGIM2003), 2003: p. 244-249.

[45] Lam, L., S.-W. Lee, and C.Y. Suen, Thinning methodologies-a comprehensive survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1992. **14**(9): p. 869-885.

[46] Van De Ville, D., W. Philips, and I. Lemahieu, Leastsquares spline resampling to a hexagonal lattice. Signal Processing: Image Communication, 2002. **17**(5): p. 393-408.

[47] Van De Ville, D., et al., Image resampling between orthogonal and hexagonal lattices. Image Processing. 2002. Proceedings. 2002 International Conference on, 2002. **3**: p. III-389-III-392 vol.3.