CHAPTER 1

DATA STRUCTURES IN IMAGE PROCESSING

1.1 INTRODUCTION

In Digital image processing the manipulation of images by computer is a relatively modern growth of human ancient captivation with visual inspirations. The essential subjective influence of pictorial displays attracts a disproportionate amount of concentration from scientist to the end use.

Image processing suffers from myths, misunderstanding, misconceptions, and misinformation. The main theme of image compression and image transmission is to reduce the storage requirement for digital imaging and also to reduce the time required for image transfer over long distances, but at the same time for reduction time in the cost of compression and the decompression time. Image restoration is the process of assessing the clean and original image by removing noise in an image where it may occur in various forms like motion blur, noise by disturbances, and camera mis-focus.

In this chapter data structures that use pixel analysis in image processing is discussed, image representation and image digitization for image pixel analysis are also explained. Here various types of data structures like traditional, hierarchical etc.., that impulse on image compression pixel data are shown.

Image segmentation is the process of connecting sets of pixels by partitioning a digital image into disjoint, one of which corresponds to the background and the remainder to the objects in the image. Segmentation matching can be used for locating objects of known appearance in an image to search for specific pattern.

Image processing is a vital supervision which reduces diverse aspects of optics, electronics, mathematics, photography, and computer technology. It is plagued with check and contradictory jargons taken from different fields. Digital image processing involves image acquisition, enhancement, restoration, compression, segmentation, representation and description.

Image acquisition is the first process in digital image processing and can be broadly explained as the action of retrieving an image from some source, usually a hardware-based source. The aim of image enhancement is to enhance the interpretability or information perception for human observers in images or to offer superior input for other automated image processing techniques.

In image processing, image compression is a vital umbrella for feature extraction. The main objective of image compression is to remove irrelevant and redundancy pixels in the image. Image compression plays an important role in medical science. Medical images need a vast hard disk space and transmission bandwidth to transmit the image. Image compression technique is a process of encoding information using less number of bits than an un-encoded representation would employ through use of specific encoding techniques.

Compression is useful to reduce the consumption of exclusive resources, such as hard disk space or transmission bandwidth. On the downside, compressed image must be decompressed and this additional processing may lead to degradation of features in the image. For occurrence, a compression scheme for image need exclusive hardware for the image to be image data compression techniques therefore contains trade-offs among several factors, among the degree of compression, the amount of distortion presented and the computational resources need to compress and decompress the data. Data compression is the manner of converting data files into less ones for efficiency of storage and transmission.

1.2 IMAGE REPRESENTATION

Image processing systems, involve mathematical characterization of image. The mathematical characterizations are: deterministic and statistical. In deterministic mathematical image function is well defined and point properties are considered. In statistical image representation, the average properties of image are explained. C(x,y,t,λ) be the spatial energy distribution (x,y)is spatial coordinates, where 't' time and λ for wavelength. Since light intensity is real positive, it is assumed that

$$0 \leq C(x, y, t, \lambda) \leq A$$

Where A is the maximum image intensity.

1.2.1 IMAGEDIGITIZATION

An image to be processed by computer will be in discrete data structure form as in a matrix. Image digitization is the function of f(x,y), sampled into a matrix of M x N. Image quantization will be done for each continuous sample an integer value. A higher the sampling rate and quantization helps in getting a good image.

There is a relationship between the density of digital sampling and the content of detail in the image. These theoretical aspects (Shannon's theorem) encourage the reader to understand at least the implications of this important result. For now, it is sufficient to appreciate that quality comparable to an ordinary television image needs, sampling into a 512 x512 grid (768 x 576 for PAL-Phase Alternating Lines format and 640 x 480 for NTSC-National Television Standard Committee format using a rectangular capture window); this is the reason why most image digitizers use this resolution. Such a resolution turns out to be adequate for a very wide range of practically useful tasks.

An image is digitized at sampling points, which are in grid. The structure image in grid structure is usually a matrix. Grids used in practice are usually square. A small sampling point in the grid structure is one picture element also called a pixel or image element in the digital image. An image element in a three dimensional is called a voxel (volume element). The set of pixels together covers the entire image; with further divisibility feature.

The transition between continuous values of the image function and its digital equivalent is called quantization. Quantization is known for high levels and quality increases in the image. Eight bits per pixel per channel (one each for red, green, blue) are commonly used although systems using other numbers (4, 6, 12 ...) can be found.

The major problem in a quantized image is the inadequacy of brightness levels with the occurrence of false contours. This arises due to the number of brightness levels are lower than what the humans can easily comprehend. Brightness level is dependent on the average local brightness but displays which avoid this effect will normally provide a range of at least 100 intensity levels. An efficient computer would represent, the brightness values in digital images in eight bits, four bits, or one bit per pixel, meaning for this is that pixel brightness one, two, or eight can be stored in one byte.

The demonstration effect of reducing the number of brightness levels in an image is explained below. An original Image with 256 brightness levels has its number of brightness levels reduced to 64, with no degradation is perceived. 16 brightness levels and false contours begin to emerge. This becomes clearer with four levels of brightness.

1.2.2 DATA STRUCTURES FOR IMAGE ANALYSIS

In the programming process, an algorithm and data are basic constraints for wiring a program and implemented in required application. Data organization affects implementation and selection of an algorithm. The data structure seems to be effective programming in images. Therefore, there are many fundamental doubts arises in writing algorithms and in programming.[Wirth, 1976]. Information about the representation of image data, and the data which can be deduced from them, will be introduced here before explaining the different image processing methods. Relations between different types of representations of image data will then be clearer. The basic level of representation of information in image analysis tasks is first dealt with and then the traditional data structures such as matrices, chains, and relational structures. Lastly we consider the hierarchical data structures such as pyramids and quadtrees.

1.2.3 LEVELS OF IMAGE DATA REPRESENTATION

The aim of computer visual perception is to find a relation between an input image and models of the real world. During the transition from the raw input image to the model, image information becomes denser with increasing use of semantic knowledge about the interpretation of image data and more. Several levels of visual information representation are defined on the way between the input image and the model; computer vision then comprises a design of the:

(1) Intermediate representations (data structures).

(2) Algorithms used for the creation of representations and introduction of relations between them.

The representations can be stratified into four levels [Ballard and Brown, 1982]-However, there are no strict borders between them and a more detailed classification of the representational levels is used in some applications. These four representational levels are ordered from signals at a low level of abstraction to the description that a human can perceive. The information flow between the levels may be bi-directional and some representations can be omitted for some specific uses.

In the first level, the lowest representational level-iconic images - consists of images containing original data: integer matrices with data about pixel brightness. Images of this kind are also the output of pre-processing operations (e.g., filtration or edge sharpening) used for highlighting some aspects of the image, which is important for further treatment.

In the second level of representation in segmentation process, images are segmented into several parts for image transmission. Such segmented parts are groups belong to same object. For instance, the segmentation is either two dimensional regions of line segments coinciding with borders corresponding to the faces of bodies. The domain deals with problems associated with image data like noise and blur that are useful. The third representational level consisting of geometric representations holding knowledge of 2D and 3D shapes. The quantification of a shape is very difficult and very important too. Geometric representations are useful while doing general and complex simulations for the influence of illumination and motion in real objects. We need them also for the transition between natural, raster images (gained, for example, by a TV camera) and data used in computer graphics (CAD – Computer-Aided Design, DTP–desktop publishing).

The fourth level of representation of image data consists of relational models. They are able to treat data more efficiently and at a higher level of abstraction. A prior knowledge about the case being solved is usually used in processing of this kind of techniques are often explored; the information gained from the image may be represented by semantic nets or frames [Nilsson, 1982].

An example will illustrate prior knowledge. Imagine a satellite image of a piece of land, and the task of counting planes standing at an airport; prior knowledge is the position of the airport, which can be deduced, for instance, from a map. Relations to other objects in the image may help as well, e.g., to roads, lakes, or urban areas. Additional prior knowledge is given by geometric models of planes for which we are searching.

1.3 TRADITIONAL IMAGE DATA STRUCTURES

Traditional image data structures such as matrices, chains, graphs, lists of object properties, and relational databases are important not only for the direct representation of image information, but also as a basis for more complex hierarchical methods of image representation.

A matrix is the most common data structure for low-level representation of an image when using devices like image capturing, the devices processing an image too directly without taking an image processing algorithms. So, the elements of matrix and corresponding sampling are processed by another characteristics and brightness. Pixels of both hexagonal and rectangular sampling grids, represented by a matrix. The relation between data and matrix elements is obvious for a rectangular grid; with a hexagonal grid every even row in the image is shifted half a pixel to the right.

Image information in the matrix is accessible through the co-ordinates of a pixel that correspond with row and column indices. The matrix is a full representation of the image, independent of the contents of image data. The image data implicitly contains spatial relations among semantically important parts of the image. The space is two-dimensional in the case of an image in a plane. One very natural spatial relation is the neighborhood one.

The segmentation in an image is obtained as a parameter for effective processing of data, when matrix processing a segmented image saves memory, explicit of all spatial relations between objects. A binary image mainly concentrates on brightness levels, segmented in zeros and ones. Each of these matrices contains spectral bond and matrices obtain hierarchical image with different resolutions. This structure is convenient for processor architectures which use parallel computers and most modern machines provide adequate physical memory to accommodate image data structures. If they do not, they are provided with virtual memory to make storage transparent. Historically, memory limitations were a significant obstacle to image applications, requiring individual image parts to be retrieved from disk independently.

When the size of the image data in the matrix is large, processing takes a long time. Algorithms can be speeded up if global information is derived from the original image matrix. Global information is concise and occupies less memory. The most popular example of global information is the histogram. Looking at the image from a probabilistic point of view, the normalized histogram can estimate the probability density of a phenomenon: that an image pixel has certain brightness.

Another example of global information is the co-occurrence matrix [Pavlidis , 1982] , which represents an estimate of the probability of two pixels appearing in a spatial relationship in which a pixel (i_1 , j_1) has intensity z and a pixel (i_2 ,j_2) has intensity y. The probability assuming that depends only on a certain spatial relation between a pixel of brightness z and a pixel of brightness y; information on the relation T is recorded in the square co-occurrence matrix Cr, whose dimensions correspond to the number of brightness levels of the image. To reduce the number of matrices Cr, some simplifying assumptions are introduced; first considering only direct neighbours first, and then treating relations as symmetrical.

Chains are used for the description of object borders in computer vision. One element of the chain is a basic symbol: This approach permits the application of a formal language theory for computer vision tasks. Chains are appropriate for data that can be arranged as a sequence of symbols. The neighboring symbols are in a chain usually corresponding to the neighborhood of primitives in the image. The primitive is the basic descriptive element that is used in syntactic pattern recognition.

This rule of proximity (neighborhood) of symbols and primitives has exceptions as for example, the first and the last symbol of the chain describing a closed border are not neighbors, but the corresponding primitives in the image are similar inconsistency and typical of image description languages [Shaw, 19691, too. Chains are linear structures, which is why they cannot describe spatial relations in the image on the basis of neighborhood or proximity.

Chain codes and Freeman codes [Freeman, 1961] are often used for the description of object borders, or other one-pixel-wide lines in images. The border is defined by the traditional image data structures, co-ordinates of its reference pixel and the sequence of symbols corresponding to the line of the unit length in several pre-defined orientations. A chain code is of a relative nature; data are expressed with respect to some reference point. An example of a chain code, is a situation where 8-neighbourhoods are used· it is possible to define chain codes using 4-neighbourhoods as well. An algorithm to extract a chain code may be implemented as an obvious simplification of Algorithm;

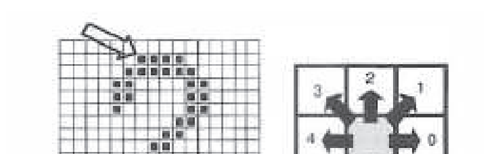

Figure 1.1: An example of chain code; The reference pixel starting the chain is marked.

When local information is needed from the chain code, it is necessary to search through the whole chain systematically. For instance, if whether the border turns somewhere to the left by 90^0, we must just find a sample pair of symbols in the chain, it is simple. On the other hand, a question about the shape of the border near the pixel (i_o, j_o) is not trivial. It is necessary to investigate all chains until the pixel(i_o, j_o) is found and only then analysis can be initiated on a short part of the border that isclose to the pixel (i_o, j_o).

The description of an image by chain is appropriate for syntactic pattern recognition that is based on formal language theory methods. When working with real images, the problem of dealing with uncertainty caused by noise arises, which explains the occurrence of several syntactic analysis techniques with deformation correction have arisen [Lu and Fu, 1978].Another method for dealing with noise is to smooth the border or to approximate it by another curve. This new border curve is then described by chain codes [Pavlidis, 1977].

Run length coding is quite often used to represent strings of symbols in an image matrix (for instance, FAX machines use full length coding). For simplicity, consider a binary image first, Run length coding records only areas that belong to the object in the image; The area is then represented as a list of lists. Various schemes exist which differ in detail - a representative one describes each row of the image by a sublist, the first element of which is the row number. Subsequent terms are co-ordinate pairs; the first element of a pair being the beginning of the second as the end (the beginning and the end are described by column coordinates). There can be several such sequences in the row. Run length coding is illustrated in Figure 1.2. The main advantage of run length coding is the existence of simple algorithms for intersections and unions of regions in the image.

Run length coding can be used for an image with multiple brightness levels as well in this case sequences of neighboring pixels in arrow that has constant brightness are considered. In the sublist we must record not only the beginning and the end of the sequence, but its brightness, too.

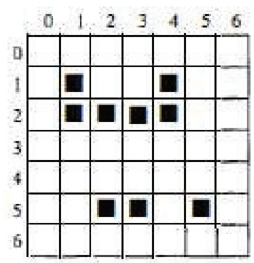

Figure 1.2: Run length coding; The code is ((1 1 144) (214) (52355)).

From the implementation point of view, chains can be represented using static data structures (e.g., 1D arrays); their size is the longest length of the chain expected. This might be too memory consuming, and so dynamic data structures are more advantageous. Lists from the LISP language are the examples.

1.4 TOPOLOGICAL DATA STRUCTURES

Topological data structures describe the image as a set of elements and relations between them which are using graph for representation. A graph G = (V, E) is an algebraic structure which consists of a set of nodes V = {v_1 ' v_2, . . . ,vn} and a set of arcs E = {e 1 ' e2 , . . . , em }.Each arc e_k is incident to an unordered (or ordered) pair of nodes {Vi ,Vj } which are not necessarily distinct [Even, 1 979] .The degree of node aspect to the number of incident arcs. A weighted graph evaluates values assigned to arcs and nodes to both these values may represent cost and weight. The region adjacent graph is clean of data structures, corresponds to neighboring and regions connected by arc. The region consists of properties like texture, color, brightness etc., to fulfill the neighborhood relations and common border to expose the segmented image.

An example of an image with areas labeled by numbers and the corresponding region adjacency graph is shown in Figure 1.3; the label 0 denotes pixels out of the image. This value is used to indicate regions that touch borders of the image in the region adjacency graph

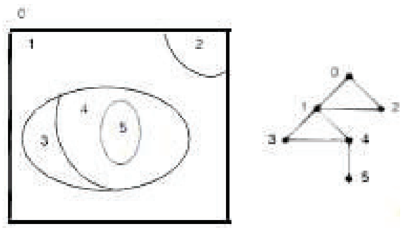

Figure 1.3: An example of region adjacency graph.

The region adjacency graph has several attractive features. If a region encloses other regions, then the part of the graph corresponding with the areas inside can be separated by a cut in the graph. Nodes of degree 1 represent simple holes. Arcs of the region adjacency graph can include a description of the relations between neighboring regions-the relations to be to the left or to be inside are common. The region adjacency graph is used for matching with the pattern for recognition purposes. The original image matrixes as dimensions are identify whose elements are labels to regions. Creation of the needs tracking of all regions is traced and labels with borders to all neighboring regions stored. The region adjutancy graph is created from the representation of images for matrix and segmentation, to the pixels parameter with neighbors a quad tree as well.

The region adjacency graph stores information about the neighbors of all regions in the image explicitly. The region map contains this information as well, but it is much more difficult to recall from there. When a quick relation of the region adjacency graph to the region map is required, making of a node in the region graph by the identification label of the region and some representation pixels (e.g the top left pixel of the region) is all that is required. Construction of the boundary data structures that represent regions is not trivial. Region adjacency graphs can be used for approaching region merging (where, for instance, neighboring regions thought to have the same image interpretation are merged into one region). In particular, intricacy of merging representations of regions that may border each other more than once is to be noted, for example, with the creation of 'holes' not present before the merge-see figure (1.4)

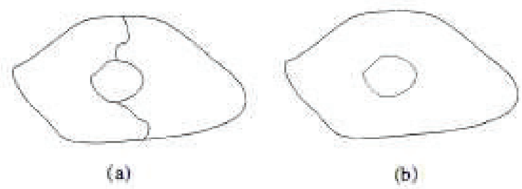

Figure 1.4: Region merging may create holes: (a) Before a merge; (b) After.

1.4.1 Relational Structures

Relational databases [Kunii et al., 1974] can also be used for representation of information from an image. The entire information is then concentrated in relations between semantically important parts of the image-objects-that are the result of segmentation. Relations are recorded in the form of tables. An example of such a representation is shown in Figure 1.4 and Table 1.1, where individual objects are associated with the images and other features, e.g., the top-left pixel of the corresponding region in the image.

Relations between objects are expressed in the relational table 1.1. In Figure 1.5, such a relation is found inside; for example, the object 7 (pond) is situated inside the object 6 (hill).Description by means of relational structures is appropriate for higher levels of image understanding. In this case searches using keys, similar to database searches, can be used to speed up the whole process.

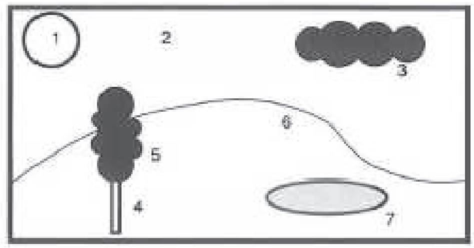

Figure 1.5: Description of objects using relational structures.

Table 1.1: Relational table.

S.No	Object Name	Color	Min.Row	Min.Col	Inside
1	Sun	White	5	40	2
2	Sky	Blue	0	0	--
3	Cloud	Gray	20	180	2
4	Tree trunk	Brown	95	75	6
5	Tree crown	Green	53	63	--
6	Hill	Light green	97	0	--
7	Pond	Blue	100	160	6

1.4.2 HIERARCHICAL DATA STRUCTURES

Computer vision is by its nature known for being expensive, if for no other reason than the large amount or data to be processed. Systems which may be called sophisticated shows process considerable quantities of image data. Usually a very quick response is expected due to the desirability of interactive systems. One of the solutions for parallel computers among processors. Many vision problems are difficult among processors or decompress in any way to programming the processes in formatting structure. Hierarchical data structures make it possible relatively to the small data

One of the solutions is the use of parallel computers among processors. Visions of problems are difficult among processors or decompress in any way to programming the processors in formatting structure. Hierarchical data structures make this possible, relatively to the small data. They achieve the finest and purpose resolution for parts of image which is essential, to speed up. The processing we introduces two structures, quad trees and pyramids for essential image processing.

Pyramids are among the simplest hierarchical data structures. Distinguishing is made between M-pyramids (matrix-pyramids) and T-pyramids (tree-pyramids). A Matrix-pyramid (M-pyramid) is a sequence $\{M_L, M_{L\_I}, \ldots \bullet, Mo\}$ of images, where M_L has the same dimensions and elements as the original image, and M_{i-1} is derived from the M_i by reducing the resolution by one-half. When creating a pyramid, it is customary to work with square matrices having dimensions equal to powers of 2- -then Mo corresponds to one pixel only.

M-pyramids are used when it is necessary to work with an image at different resolutions simultaneously. An image with one degree smaller resolution in pyramid contains four times less data, so it is processed approximately four times as quickly. It is simultaneous use of several resolutions simultaneously rather than choosing just one image from the M-pyramid is advantageous in many cases. For such algorithms use of tree-pyramids is preferable to a tree structure. Let 2^L be the size of an original image (the highest resolution). A tree pyramid(T-pyramid) is defined by:

1. A set of nodes $P = \{ p = (k,i,j)$ such that level $k \in [0, L]$; $i, j \in [0,2^k - 1]\}$.
2. A mapping F between subsequent nodes P_{k-1}, P_k of the pyramid

F (k,i,j) = (k - 1 , i div 2 ,j div 2) ,Where 'div' denotes whole-number division.

3. A function V that maps a node of the pyramid P to Z, where Z is the subset of the whole numbers corresponding to the number of brightness levels, for example, Z = {0, 1 , 2, . . . , 255} .

Nodes of T-pyramid correspond for a given k with image points of an M-pyramid; elements of the set of nodes P = {(k, i, j)} correspond with individual matrices in the M-pyramid-k is called the level of the pyramid. An image P = {(k, 'i, j)} for a specific k constitutes an image at the 'k'th level of the pyramid. F is the so-called parent mapping, which is defined for all nodes P_k of the T-pyramid except its root (0, 0, 0). Every node of the T-pyramid has four child nodes except leaf nodes, which are nodes of level L that correspond to the individual pixels in the image.

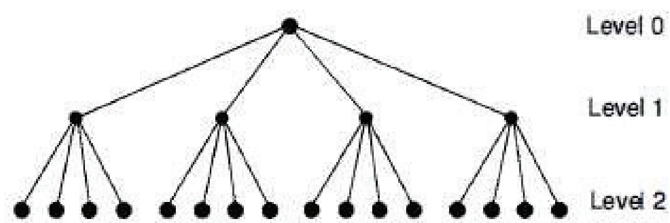

Figure 1.6: T-pyramid.

Values of individual nodes of the T-pyramid are defined by the function V. Values of leaf nodes are the same as those of the image function (brightness) in the original image at the finest resolution with the image size as 2^{L-1}. Each value of node in other levels of the tree are either an arithmetic mean of four child nodes or they are defined by coaster sampling, meaning that the value of one child (e.g., top left) is used. Figure 1.6 shows the structure of a simple T-pyramid. The number of image pixels used by an M-pyramid for storing all matrices is given by

$$N2\left(1+\left(\frac{1}{4}\right)+\left(\frac{1}{16}\right)+\cdots.\right)\approx 1.33N2$$

Where N is the dimension of the original matrix (the image of finest resolution)-usually a power of two, 2^L.

There is a similar representation of the T pyramid in memory. Arcs of the tree need not be recorded because addresses of the both child and parent nodes are easy to compute due to the regularity of the structure. An algorithm for the effective creation and storing of a T-pyramid is given by Pavlidis, 1982.

Quadtrees are implemented from hierarchical data structures to processing the parallel computation in image processing. These trees are modification of t-pyramids. In this all nodes except the leaves (NW, North-Western; NE, North-Eastern; SW, South-Western; SE, South Eastern). Like T-pyramids, the image is divide into four quadrants, in hierarchical levels. The levels and quadrants are shows brightness, it is not fit to record them. This representation is less expensive for an image with large homogeneous regions;

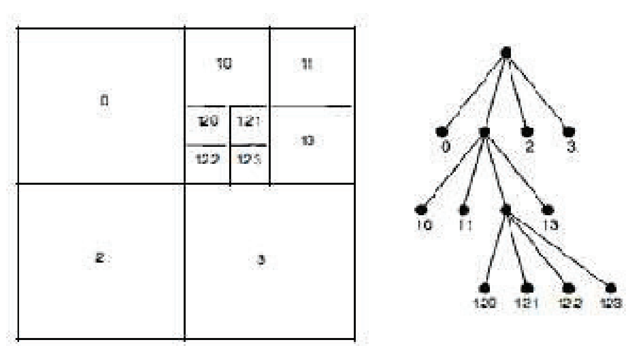

Figure 1.7 is an example of a simple quad tree.

An advantage of image representation by means of quad trees is the existence of simple algorithms for addition of images, computing object areas, and statistical moments. The main disadvantage of quad trees and pyramid hierarchical representations is their dependence on the position, orientation, and relative size of objects. Two similar images with just very small differences can have very different pyramid or quad tree representations. Even two images depicting the same, with slight shifting of scene, can have entirely different representations.

These disadvantages can be overcome using a normalized shape of quad tree wherein create the quad tree for the whole image is not created but, for its individual objects. Geometric features of objects such as the center of gravity and principal axis are used. The center of gravity and principal axis of every object are derived first and then the smallest enclosing square centered at the center of gravity having sides parallel with the principal axes is located. The square is then represented by a quad tree. An object described by a normalized quad tree and several additional items of data (co-ordinates of the center of gravity, angle of main axes) is invariant to shifting, rotation, and scale.

Quad trees are usually represented by recording a whole tree as a list of its individual nodes, every node being a record with several items characterizing it. An example is given in table1.2. In the item Node type there is information on whether the node is a leaf or inside the tree. Other data can be the level of the node in the tree, position in the picture, code of the node, etc. This kind of representation is expensive in memory. Its advantage is easy access to any node due to presence of pointers between parents and children.

It is possible to represent a quad tree with smaller demand on memory by means of a leaf code. Any point of the picture is coded by a sequence of digits reflecting successive divisions of the quad tree; zero means the NW (north-west) quadrant, and likewise for the other quadrants: 1-NE, 2-SW, 3-SE. The most important digit of the code (on the left) corresponds to the division at the highest level, the least important one (on the right) with the last division. The number of digits in the code is the same as the number of levels of the quad tree.

Table 1.2: Record describing a quad tree node.

Node type
Pointer to the NW son
Pointer to the NE son
Pointer to the SW son
Pointer to the SE son
Pointer to the father
Other data

The whole tree is then described by a sequence of pairs of leaf code and the brightness of the region. Programs creating quad trees can use recursive procedures to advantage.

T-pyramids are very similar to quad trees, but differ in two basic respects. A T-pyramid is a balanced structure, that means the corresponding tree divides the image regardless of the contents, which is is regular and symmetric. A quad tree is not balanced. The other difference is in the interpretation of values of the individual nodes.

Quad trees have seen widespread application, particularly in the area of Geographic Information Systems (GIS) where, along with their three-dimensional generalization octrees, they have proved very useful in hierarchical representation of layered data [Samet, 1989, 1990] .

The pyramidal structure is widely used, and has seen several extensions and modifications. Recalling that a (simple) M-pyramid was defined as a sequence of images $\{M_L, M_{L-1} \ldots M_0\}$ in which M_i is a 2 x 2 reduction of M_{i+1} , we can define the conception of a reduction window can be defined. For every cell c of M_i , the reduction window is its set of children in M_{i+1}, w (c). Here, a cell is any single element of the image M_i at the corresponding level of pyramidal resolution.

If the images are constructed with all interior cells having the same number of neighbors (e.g., a square grid, as is customary), and with all of them having the same number of children, the pyramid is called regular. Taxonomy of regular pyramids may be constructed by considering the reduction window together with the reduction factor λ, which defines the rate at which the image area decreases between levels;

$$\lambda \leq \frac{|Mi+1|}{|Mi|}, i = 0,1,2,3 \ldots\ldots, L-1.$$

In the simple case, in which reduction windows do not overlap and are 2 x 2. The factor is reduced when overlapping of reduction windows is taken up. The notation used to describe this characterization of regular pyramids is (reduction window)/(reduction factor). Figure 1.8 illustrates some simple examples.

The reduction window of a given cell at level i may be propagated down to higher resolution than level i + 1. For a cell C_i at level i, we can write $W^0(C_i) = W(C_i)$, and then recursively define $w^{k=1}(c_i)$=U $w^k(q)$ where q⊑ $w(ci)$$w^k(C_i)$ is the equivalent window that covers all cells at level i+k+1 that link to the cell C_i . Note that the shape of this window is going to depend on the type of pyramid for example, an n x n/2 pyramid will generate octagonal equivalent windows, while for an n x n/4 pyramid they will be square. Use of non-square windows prevents domination of square features, as is the case for simple 2 x 2/4 pyramids.

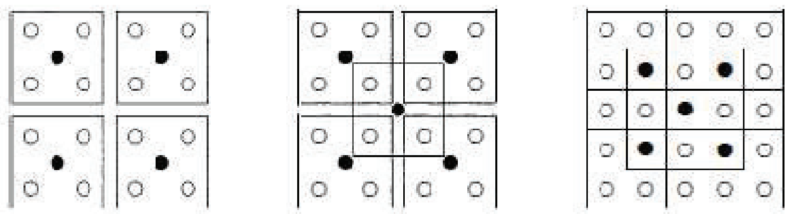

Figure 1.8: Several regular pyramid definitions. (a) 2 x 2/4. (b) 2 x 2/2. (c) 3 x 3/2. (Solid dots are at the higher level, ie., the lower-resolution level.)

The 2 x 2/4 pyramid i s widely used and is what is usually called an 'image pyramid'; the 2 x 2/2 structure is often referred to as an 'overlap pyramid'. 5 x 5/2 pyramids have been used [Burt and Adelson, 1983] in compact image coding, where the image pyramid is augmented by a Laplacian pyramid of differences.

Here, the Laplacian pyramid at a given level is computed as the per-pixel difference between the image at that level, and the image derived by 'expanding' the image at the next lower resolution. The Laplacian may be expected to have zero (or close) values in areas of low contrast, and therefore be an enable to compression.

Irregular pyramids are derived from contractions of graphical representations of images (for example, region adjacency graphs). Here, a graph may be reduced to a smaller one by selective removal of arcs and nodes. Depending on how these selections are made, important structures in the parent graph may be retained while reducing its overall complexity [Kropatsch, 1995].

The pyramid approach is quite general and lends itself to many developments for example, the reduction algorithms need not be deterministic [Meer,1989]. A brief survey of pyramidal segmentation algorithms may be found in [Bister et al., 1990].

1.5 CONCLUSION

In this chapter, the purpose of image compression and their usage in various applications of image processing is discussed with the analysis of data structures for image compression and transmission with the use of these methods of pixel data structures is also explained.