2
Mathematical Preliminaries

In this chapter, a brief overview of mathematical preliminaries have been explained which are used in the various parts of the thesis. These preliminaries consist of scale-space transformations, matrix decomposition and feature estimation techniques. The details of these preliminaries are as follows.

2.1 DISCRETE WAVELET TRANSFORM (DWT)

Fourier analysis is an widely used tool in literature for analysing different types of 1D/2D signals but still in some cases Fourier analysis fails to provide better description of a signal having low frequencies of long duration and high frequencies of short duration. Wavelet analysis can analyse these type of signals due to its ability of having large time-resolution for short-lived high-frequency and has large frequency-resolution for long-lived low-frequency. It is generally based on a multiresolution analysis (MRA). Mathematically, a multiresolution analysis (MRA) with l levels of a signal $f(x) \in L^2(R)$ is a projection of f on a basis $\{\phi_{j,k}, \{\psi_{j,k}\}_{j\leq l}\}_{k\in Z}$. Basis function $\phi_{j,k}(x) = 2^{-j/2}\phi(2^j x - k)$ results from translation and dilation of a function $\phi(x)$ called scaling function. The family $\{\phi_{j,k}\}_{k\in Z}$ spans a subspace $V_j \subset L^2(R)$. The projection of f on V_j gives an approximation $\{a_{j,k} =< f, \phi_{j,k} >\}_{k\in Z}$ of f at the scale 2^j. Analogously, function $\psi_{j,k}(x) = 2^{-j/2}\psi(2^j x - k)$ results from translation and dilation of a function $\psi(x)$ called mother wavelet function. The family $\{\psi_{j,k}\}_{k\in Z}$ spans a subspace $W_j \subset L^2(R)$. The projection of f on W_j gives an approximation $\{w_{j,k} =< f, \psi_{j,k} >\}_{k\in Z}$ of f representing the *details* between two successive approximations. The MRA with l levels yields the following decomposition of f

$$f(x) = \sum_k a_{j,k}\ \tilde{\phi}_{j,k}(x) + \sum_{j\leq l}\sum_k w_{j,k}\ \tilde{\psi}_{j,k}(x) \tag{2.1}$$

Due to the separability of the transform, MRA of an image can be obtained by successively taking 1D MRA along both the directions. The MRA of an image yields four components: approximation, horizontal, vertical and diagonal coefficients [Mallat, 1989]. The approximate coefficients capture the low frequency components of the image whereas the horizontal, vertical and diagonal coefficients capture the high frequency components in varied directions and are called detail coefficients. These components can be efficiently constructed based on lifting scheme. The basic principle of lifting scheme is based on the improvement wavelet and its dual to meet the basic requirement of the practical applications, with maintaining the bi-orthogonality of the wavelets. In addition, the lifting wavelet transform preserves the better spatial and spectral localization in comparison to the traditional wavelet transform [Soman, 2010]. The lifting wavelet transform comprises three steps namely split, predict and update. These steps can be described as given below [Sweldens, 1996]:

- **Split:** The input signal $z(n)$ is split into two usual components with no common elements. These components can be represented by even series $Z_e(n)$ and odd series $Z_o(n)$.

$$Z_e(n) = Z(2n), \tag{2.2}$$

$$Z_o(n) = Z(2n+1),\ \ s.t.\ Z_e(n) \cap Z_o(n) = \phi. \tag{2.3}$$

- **Predict:** Both the samples Z_e and Z_o are created by splitting the signal, so there exists a close correlation between them. Hence, odd samples can be predicted by keeping unchanged the even samples. Firstly, a predicted operator P is applied on the even samples $Z_e(n)$ and then the difference between the predicted values $P[Z_e(n)]$ and $Z_o(n)$ turns out results as the detail signal. Mathematically,

$$d(n) = Z_o(n) - P[Z_e(n)] \tag{2.4}$$

where predictor operator (P) modifies the high-frequencies and represents the wavelet coefficients in $d(n)$.

- **Update:** An update operator U is finally applied on the signal $d(n)$ and then even sample $Z_e(n)$ is modified by the updated detail signal $U[d(n)]$. The update process is given as follows:

$$c(n) = Z_e(n) + U[d(n)] \tag{2.5}$$

where $c(n)$ corresponds to the low-frequency component of the original signal.

One lifting process of the considered signal can be accomplished by the above three steps. The complete lifting scheme is illustrated in Fig. 2.1. In contrast, for visual perception, the decomposition of cameraman image corresponding based on lifting scheme is shown in Fig. 2.2.

2.2 INTEGER DISCRETE COSINE TRANSFORM (IDCT)

In many of the practical applications, the input sources are in the form of images or videos, which can be generally represented by the integer-data sequences. This property of the input can be viewed as a constraint to an integral sub-lattice of the input space and when these signals or images are projected by an arbitrary linear transformation for further processing and analysis, it is by and large impossible to make this constraint practically tractable. In other words, the outputs generally do not belong to the same sub-lattice. For example, the Fourier transform will project the integer sub-lattice onto the complex sub-lattice. In applications where the inversion is required, it is important to have a simple and fast reversal process, which is not possible when the output sub-lattice is different than the input sub-lattice. Therefore, a linear transformation is essential which maps integer input to integer output so that the inversion is simple and fast. These transformations are called integer linear transformation. Up to now, many integer linear transformations have been proposed for image and video coding applications. These are based on the lifting factorization of the standard Fourier and Cosine transform matrices [Oraintara *et al.*, 2002; Pei and Ding, 2000;

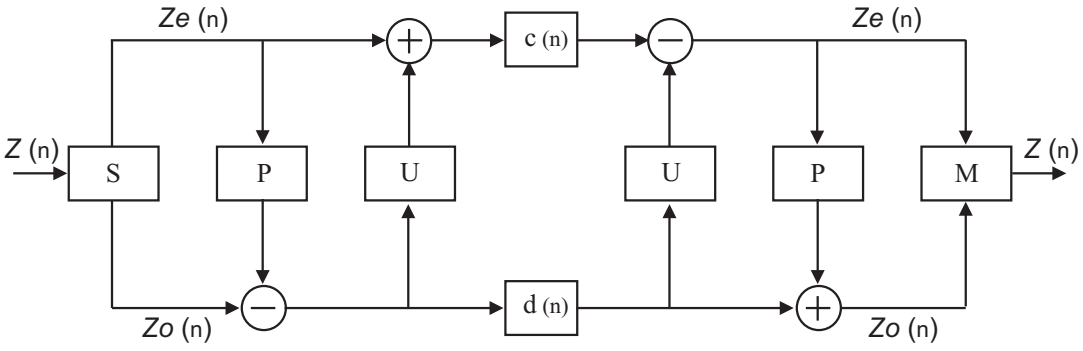

Figure 2.1 : Lifting wavelet decomposition and reconstruction

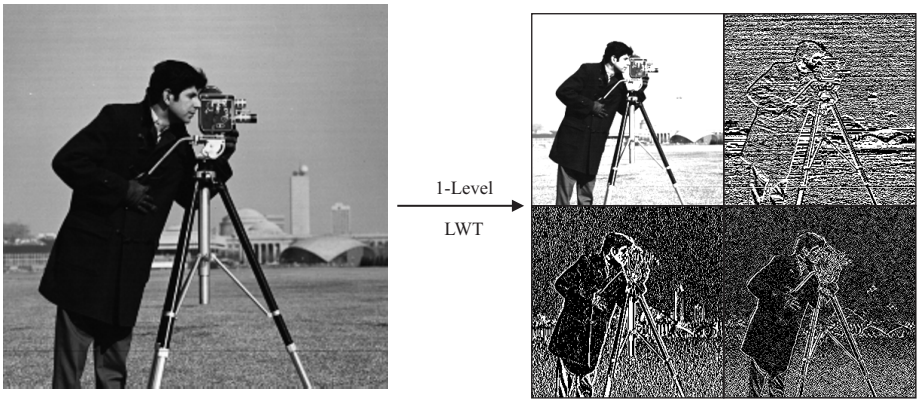

Figure 2.2 : First-level lifting wavelet decomposition of an image and corresponding different sub-band.

Plonka and Tasche, 2003; Suzuki and Ikehara, 2010; Zeng *et al.*, 2001] . In this work, integer-DCT is used as the integer linear transformation mainly due to its high ability of data decorrelation [Zeng *et al.*, 2001]. Mathematically, the two dimensional integer-DCT (2D-IntDCT) of an image $x = \{x(n,m) : n = 0,1,...,N-1; m = 0,1,...M-1\}$ can be computed as per the following steps [Zeng *et al.*, 2001].

1. Compute $y(p,m) = x(q(p,m),m)$, where

$$q(p,m) = \begin{cases} 2f(p,\frac{m}{2}), & m \text{ is even and } f(p,\frac{m}{2}) < \frac{N}{2} \\ 2N-1-2f(p,\frac{m}{2}), & m \text{ is even and } f(p,\frac{m}{2}) \geq \frac{N}{2} \\ 2f(p,M-\frac{m+1}{2}), & m \text{ is odd and } f(p,M-\frac{m+1}{2}) < \frac{N}{2} \\ 2N-1-2f(p,M-\frac{m+1}{2}), & m \text{ is odd and } f(p,M-\frac{m+1}{2}) \geq \frac{N}{2} \end{cases} \tag{2.6}$$

   with $p = 0,1,...,N-1$; $m = 0,1,...,M-1$ and $f(p,m) = [(4p+1)m+p] \mod N$.

2. Apply the following steps to find the 1D intDCT of each row of $y(p,m)$:

   a) Construct the lifting matrices $\bar{F}_{\frac{M}{2}}$ and $\bar{E}_{\frac{M}{2}}$ as follows.

$$\bar{F}_{\frac{M}{2}} = \left\{ \prod_{k=0}^{\frac{M}{4}-1} \Big[L_{2k,2k+1}(RB(\alpha_{2k})-1) L_{2k+1,2k}(1) \times L_{2k,2k+1}\Big(RB\big(1/\alpha_{2k}\big)-1\Big) \right.$$

$$\left. L_{2k+1,2k}\Big(-RB(\alpha_{2k})\Big)\Big] \right\} \left\{ \prod_{k=1}^{\frac{M}{4}-1} \Big[L_{2k-1,2k}\Big(RB(\alpha_{2k-1})-1\Big) L_{2k,2k-1}(1) \right.$$

$$\left. \times L_{2k-1,2k}\Big(RB\big(1/\alpha_{2k-1}\big)-1\Big) L_{2k,2k-1}\Big(-RB(\alpha_{2k-1})\Big)\Big] \right\} \tag{2.7}$$

   and

$$\bar{E}_{\frac{M}{2}} = diag(RB(\sqrt{2}),1,...,1). \tag{2.8}$$

   where the notation $RB(s)$ is used to denote a dyadic rational number of form $\beta/2^{\lambda}$ that approximates the real number s.

b) Compute 1D-IntDCT transform matrix as

$$\bar{C}_M = P_M \begin{bmatrix} I_{\frac{M}{2}} & 0 \\ 0 & U_{\frac{M}{2}} \end{bmatrix} \begin{bmatrix} C_{\frac{M}{2}} & 0 \\ 0 & C_{\frac{M}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{M}{2}} & 0 \\ 0 & \bar{D}_{\frac{M}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{M}{2}} & \hat{I}_{\frac{M}{2}} \\ 0 & -\hat{I}_{\frac{M}{2}} \end{bmatrix} \tag{2.9}$$

where $\bar{D}_{\frac{M}{2}} = \bar{E}_{\frac{M}{2}} \bar{F}_{\frac{M}{2}}$ and

$$U_{\frac{M}{2}} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & ... & ... & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & ... & ... & 0 & 0 \\ \frac{1}{2} & -1 & 1 & ... & ... & 0 & 0 \\ & ... & & ... & ... & ... & \\ \frac{1}{2} & -1 & 1 & ... & ... & 1 & 0 \\ -\frac{1}{2} & 1 & -1 & ... & ... & -1 & 1 \end{bmatrix} \tag{2.10}$$

c) Compute the 1D IntDCT as $V(p,l) = \bar{C}_m y(p,m)$.

3. Compute a polynomial transform as

$$A_k(z) = \sum_{p=0}^{N-1} V_p(z) \hat{z}^{pk} \mod z^{2M} + 1; k = 0, 1, ..., N-1 \tag{2.11}$$

where $\hat{z} \equiv z^{2^{j+2}} \mod z^{2M} + 1$ and

$$V_p(z) = \sum_{l=0}^{M-1} V(p,l) z^l - \sum_{l=M+1}^{2M-1} V(p, 2M-l) z^l \tag{2.12}$$

and then get $B_k(z) \equiv A_k(z) z^{2^j k} \mod z^{2M} + 1$.

4. Find the 2D-IntDCT values X={$X(k,l) : k = 0, 1, ..., N-1; l = 0, 1, ..., M-1$ } from the following equation

$$X_k(z) = \sum_{l=0}^{2M-1} X(k,l) z^l \equiv \frac{1}{2}(B_k(z) + B_k(z^{-1})) \mod z^{2M} + 1 \tag{2.13}$$

The original image x can be obtained from the transformed values X using the two dimensional inverse integer-DCT (2D-InvInt-DCT). The inverse transform can be summarized in the following steps.

1. Generalize the polynomials $X_k(z) = \sum_{l=0}^{M-1} X(k,l) z^l - \sum_{l=M+1}^{2M-1} X(k, 2M-l) z^l$, and compute

$$A_k(z) \equiv (X_k(z) + X_{N-k}(z) z^{-M}) z^{k2^j} \mod (z^{2M} + 1). \tag{2.14}$$

2. Compute a polynomial transform as

$$V_p(z) \equiv \frac{1}{N} \sum_{k=0}^{N-1} A_k(z) \hat{z}^{pk} \mod z^{2M} + 1 \tag{2.15}$$

where $p = 0, 1, ..., N-1$ and the coefficients of $V_p(z)$ are denoted by $V(p,l)$.

3. Apply the steps given in eqns. (17-21), to find the 1D intDCT of each row of $V(p,l)$ and let the output is denoted by $y(p,m)$.

4. Reorder the array $y(p,m)$ to get $x(n,m) = y(r(n,m),m)$, where

$$r(n,m) = \begin{cases} g(\frac{n}{2}, \frac{m}{2}), & \text{Both } m \text{ and } n \text{ are even} \\ g(N - \frac{n+1}{2}, \frac{m}{2}), & m \text{ is even and } n \text{ is odd} \\ g(\frac{n}{2}, M - \frac{m+1}{2}), & m \text{ is odd and } n \text{ is even} \\ g(N - \frac{n+1}{2}, M - \frac{m+1}{2}), & \text{Both } m \text{ and } n \text{ are odd} \end{cases} \tag{2.16}$$

with $n = 0,1,...,N-1$; $m = 0,1,...,M-1$ and $g(n,m) = [(4m+1)^{-1}(n-m)] \mod N$.

2.3 MATRIX DECOMPOSITION METHOD

Matrix decomposition is refers to the mathematical operations that transform it into the canonical form. The main idea of matrix decomposition is to simplify the more typical operations that can be applied on the decomposed parts rather than the original matrix. This decomposition is used in many scientific applications due to its computational convenience. Some of the important matrix decomposition methods are briefly described as follows.

2.3.1 Singular Value Decomposition

A singular value decomposition (SVD) is a mathematical transformation used for the factorization of the matrices. Suppose A is a matrix of size $m \times n$ defined over the field F (where F may be real or complex) can be expressed into product of three matrices such that

$$A = USV^T \tag{2.17}$$

where U and V are the unitary matrices of size $m \times m$ and $n \times n$, S is diagonal matrix of size $m \times n$ whose diagonal elements are the singular values of A and V^T denotes the conjugate transpose of V. The columns of U and V are the left and right singular vector of A. This decomposition can be expressed as follows:

$$A = \sum_{i=1}^{r} \vec{u}_i s_i \vec{v}_i^T \tag{2.18}$$

where r is the rank of matrix A, s_i is the i^{th} singular value, $\vec{u}_i$ and $\vec{v}_i$ are the i^{th} left and right singular vectors respectively. For digital images, SVD is a technique for transforming correlated pixels into a set of uncorrelated ones that better expresses different relationship among the original pixels. In principle, singular values and singular vectors represent the brightness and the geometrical features of the image. A detailed analysis on the importance of SVD in digital image processing can be seen in [Andrews and Patterson, 1976].

2.3.2 QR Decomposition

The QR decomposition is another well-known factorization technique, which essentially factorizes a matrix A of size $m \times n$ into the following form [Golub and Van Loan, 2012]:

$$A = Q * R \tag{2.19}$$

where Q is an orthogonal matrix of size $m \times n$ and R is a upper triangular matrix of size $m \times m$. The QR decomposition can be computed using the householder transformation. This transformation corresponds to a householder matrix (H) as follows:

$$H = I_n - 2\frac{< v, v^t >}{< v^t, v >} \tag{2.20}$$

where H and I represent an orthogonal matrix and identity matrix respectively, v is a non-zero real vector. There are k steps involved in the overall procedure, where $k = min(m-1, n)$. The QR decomposition can be computed as:

$$R = (H_k H_{k-1} \dots H_2 H_1) A \tag{2.21}$$

$$\Rightarrow R = (Q)^T A \Rightarrow A = Q * R \tag{2.22}$$

2.4 CHAOTIC MAP

Chaos theory is a scientific discipline that concentrated on investigation of the behaviour of non-linear chaotic systems that are highly sensitive on its initial state [Lasota and Mackey, 2013]. Chaotic systems have many important properties, such as randomness, aperiodicity, unpredictability, sensitive dependence on initial conditions and parameters. These properties make chaotic systems an ideal choice for those applications where security is of thrust interest [Hasselblatt and Katok, 2003]. Due to usefulness of these characteristics, chaotic systems are widely used in communication and image processing applications. Some of the chaotic maps used in the proposed work are listed below.

2.4.1 Piece-wise Non-linear Chaotic Map

Chaotic systems have many important properties, such as randomness, aperiodicity, unpredictability, sensitive dependence on initial conditions and parameters. These properties make chaotic systems an ideal choice for those applications where security is of thrust interest. In this work, a non-linear chaotic system is used to generate a random sequence. Mathematically, the piece-wise non-linear chaotic map (PWNLCM) can be defined as given below:

$$\mathscr{G}(z_{k+1}) = \begin{cases} \left(\frac{1}{I_{i+1}-I_i} + \alpha_i\right)(z_k - \alpha_i) - \frac{a_i}{I_{i+1}-I_i}(z_k - \alpha_i)^2, & \text{if } z_k \in [I_i, I_{i+1}) \\ 0, & \text{if } z_k = 0.5 \\ \mathscr{G}(z_k - 0.05), & \text{if } z_k \in (0.5, 1] \end{cases} \tag{2.23}$$

where $z_k \in I = [0,1]$ with considering initial state z_0. I_i denote a subinterval of I such that $0 = I_0 < I_1 < \ldots < I_{n+1} = 0.5$. The parameter $\alpha_i \in (-1,0) \cup (0,1)$ tune sequence in the i^{th} interval such that

$$\sum_{i=0}^{n-1} (I_{i+1} - I_i)\alpha_i = 0 \tag{2.24}$$

Some of the important characteristic of chaotic system are given below.

- **Sensitivity to initial conditions:** The initial state helps to analyze the future behaviour of a chaotic system. Any slight change in the initial state results into a different outcome.
- **Mixing:** It refers to properties of a dynamical system, in which the system will evolve over the time so that any given region of states always overlaps with any other given region.

2.4.2 Arnold Transformation

Arnold transformation is a two-dimensional map which is essentially used to scramble the given image. In image scrambling process, the original pixel position is encoded by a new position based on an iterative process. Let $\hat{F} = \{(u,v) | u, v = 0, 1, 2, ..., N-1\}$ denote the original image of size $N \times N$, then the Arnold transform [Arnol'd and Avez, 1968; Sui and Gao, 2013] can be defined as follows:

$$\begin{bmatrix} u_n \\ v_n \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & 1+ab \end{bmatrix} \begin{bmatrix} u_{n-1} \\ v_{n-1} \end{bmatrix} \ mod \ \ N \tag{2.25}$$

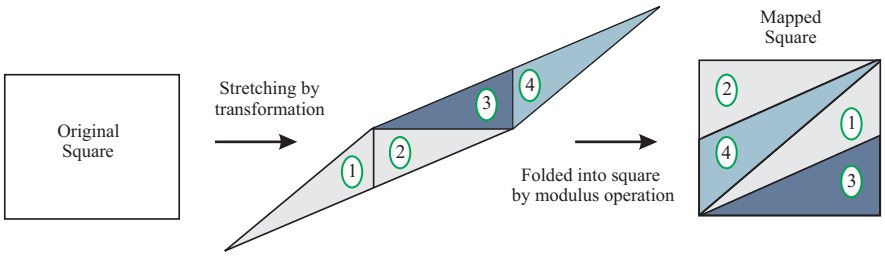

Figure 2.3 : Working process of arnold transformation

where u_n and v_n represent the transformed coefficients with respect to original values u_{n-1} and v_{n-1} after n^{th} iterations. Also, the parameters a and b are positive and belong to the set of real numbers. The Arnold transformation changes the position of (u,v) several times and after t^{th} iteration it returns to original position. The value of t is called the period of the transformation. The iterative process breaks the correlation between the image pixels and after few iterations the pixels bear no correlation in the transformed image. The working process of Arnold transformation is depicted in Fig. 2.3.

2.5 LOG-POLAR MAPPING

Log-polar mapping (LPM) can be viewed as a non-linear and non-uniform sampling of the spatial domain. The non-linearity is obtained by the polar-mapping whereas logarithmic scaling will lead to non-uniform sampling. Thus, log-polar mapping can be defined as the composition of two transformations f and g such that

$$f : C \to P \qquad g : P \to L_P \tag{2.26}$$

where C, P and L_P represent the cartesian, polar and log-polar coordinate systems respectively. The motivation for the inclusion of log-polar mapping comes from its biological genesis. In biology, LPM is an accepted model of retinal representation in the primary visual cortex. The nonuniform sampling that imitate logarithmic scale takes place in the retina, which are connected to the visual cortex by a mapping pretended by the nerves. A simple rewiring by the radial nerves, this mapping realizes the polar transformation. Due to this, LPM is referred to the retino-cortical communication. In principal, cartesian, polar and log-polar coordinate system represents the image, retinal and cortical planes, as per the human visual system (HVS).

Given an image, it is first transformed into the log-polar sampling and it is then mapped into the log-polar plane as shown in Fig. 2.4. Typically, the sampled image is divided, as per HVS, into two distinct regions called (1) fovea and (2) periphery to produce a log-polar image. The fovea constitute the interior part of the log-polar image by organizing different regions hexagonally. In contrast, the similar regions are scattered in circular fashion to constitute the periphery. The area of periphery is increased exponentially with respect to the center of the cortical plane. The complete LPM process can be summarized as follows.

Consider a point (x,y) and (ρ,θ) in cartesian and log-polar coordinate system respectively, one can have

$$\rho(x,y) = log_a \frac{\rho}{\rho_0} \text{ and } \theta(x,y) = q\phi \tag{2.27}$$

where ρ denotes the radial distance from the center of the mapping, ρ_0 is the radius of the innermost circle and $\frac{1}{q}$ is the the minimum angular resolution of the log-polar mapping. These relations

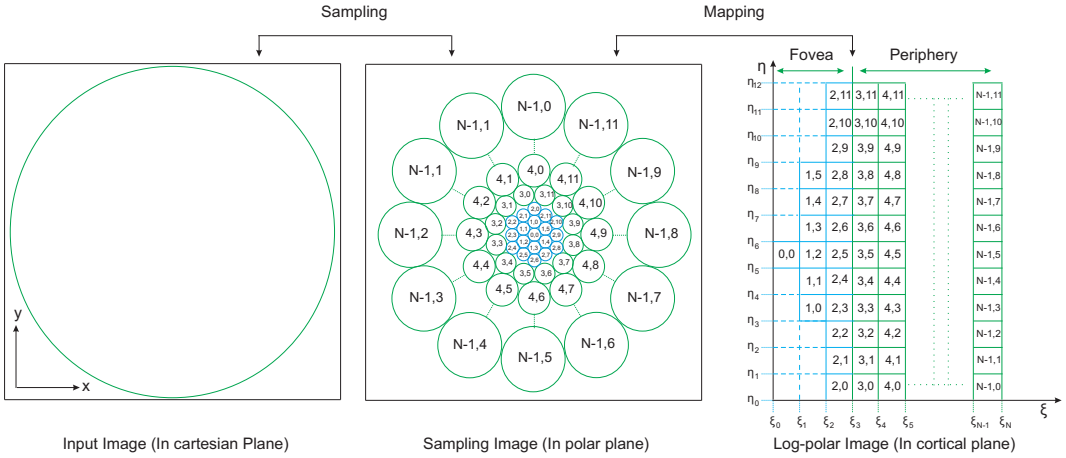

Figure 2.4 : An illustration of the log-polar mapping as defined in Eqn. 2.26.

can be referred as to conventional cartesian system and are defined as:

$$\rho = \sqrt{x^2 + y^2} \text{ and } \phi = arctan\frac{y}{x} \tag{2.28}$$

Despite the inconvenience of non-linear processing, the log-polar mapping has received substantial attention for computer vision applications mainly due to the following advantages. (1) the log-polar representation is invariant to rotation and scaling, and (2) the spatially varying sampling in the retina reduces the amount of information through the nerves while maintaining high-resolution in the fovea. This property is helpful to process a high-resolution image efficiently. A detailed description of the LPM and underlying sampling and mapping process can be found in Zheng *et al.* [2003].

2.6 DECIMAL SEQUENCE GENERATION

Decimal sequences exhibit good correlation property such as auto-correlation and cross-correlation and therefore used in many practical applications which are using pseudo-random sequences. A decimal sequence can be produced by considering a positive integer number when it is expressed in a decimal representation with respect to some base $\hat{r}$. Mathematically, a binary d-sequence can be produced with base $\hat{r} = 2$ as follows.

$$a(j) = \{2^j \bmod q\} \bmod \hat{r} \qquad j = 1, 2.....q-1, \tag{2.29}$$

where q denotes a prime number. In [Parakh, 2006], authors introduce a non-linearity in the generation process of binary d-sequence by adding more than two different binary d-sequences using a sequence of prime numbers $q_1, q_2.....q_n$ as follows.

$$a(j) = \{2^j \bmod q_1\} \bmod 2 \oplus \{2^j \bmod q_2\} \bmod 2 \oplus \{2^j \bmod q_3\} \bmod 2.... \tag{2.30}$$

where $\oplus$ denote the modular addition operation. A new decimal sequence is then generated using the recursion in the above approach as follows:

$$\begin{aligned} a(j) = &\{(s^j \bmod q_{11} + s^j \bmod q_{12} + ... + s^j \bmod q_{1n})^k \bmod q_{21}\} \bmod 2 \oplus \\ &\{(s^j \bmod q_{11} + s^j \bmod q_{12} + ... + s^j \bmod q_{1n})^k \bmod q_{22}\} \bmod 2 \oplus ..... \\ &....\{(s^j \bmod q_{11} + s^j \bmod q_{12} + ... + s^j \bmod q_{1n})^k \bmod q_{2m}\} \bmod 2. \end{aligned} \tag{2.31}$$

where s and q_{ij} are prime numbers and s represents the initial seed for the decimal sequence generation.

2.7 KAZE FEATURES

KAZE is a new feature detection and description technique that uses the non-linear scale space representation to extract the features of an image. The approach is similar to the SIFT and SURF features, however the major difference lies between the building of the scale space. KAZE features [Alcantarilla *et al.*, 2012] utilize the non-linear diffusion filter instead of Gaussian scale space. The Gaussian approach essentially blurs the noise and details in the same degree, loses the natural boundary of the object when evolving the original image through the scale space. The main advantage comes with non-linear diffusion filter that removes the noise while keeping important details of the image. The scheme is based on variable conductance diffusion and Additive Operator Splitting (AOS) method which is totally stable for any step size. The non-linear diffusion approaches can be explained using the classic partial differential equation and can be described as follows:

$$\frac{\delta F_s}{\delta t} = div\{c(x,y,t).\bigtriangledown(F_s)\} \tag{2.32}$$

where $\bigtriangledown$ and div are respective mathematical symbols used to represent the gradient and divergence operators. The function $c(\cdot)$ defines the conductivity in differential Eqn. 2.32 and can be described as follows as:

$$c(x,y,t) = g(|\bigtriangledown F_\sigma(x,y,t)|) \tag{2.33}$$

where the function $\bigtriangledown F_\sigma$ defines the gradient of the original image F having standard deviation σ. The two different mathematical expressions representing the conductivity function g given as:

$$G1 = \exp\left(-\frac{|\bigtriangledown F_\sigma|^2}{k^2}\right) \text{ and } G2 = \left(\frac{1}{1+\frac{|\bigtriangledown F_\sigma|^2}{k^2}}\right) \tag{2.34}$$

where G1 and G2 are used to represent the high contrast edge and wider regions respectively. The Eqn. 2.32 can be discretization and can be written as:

$$\frac{F^{i+1}-F^i}{\tau} = \sum_{\ell=l}^{m} A_\ell(F^i)F^{i+1} \tag{2.35}$$

where F^i and F^{i+1} are smoothed images at present and adjacent level respectively. Also, A_ℓ is the matrix that encodes the diffusivities for each image dimension. In discretization, a similar approach to the SIFT is applied which maps set of levels into octave and sub-levels with the help of octave index o and sub-level index s. The octave and the sub-level are mapped to their respective scale σ in the scale space of N_L filtered image and can be defined as follows:

$$\begin{aligned}\sigma_i(o,s) &= \sigma_0 2^{(o+s)}/S, i \in [0,...N_L-1],\\ o &\in [0,...N_{oct}-1], s \in [0,...N_{sub}-1]\end{aligned} \tag{2.36}$$

where σ_0 representing the base scale levels. Since, non-linear diffusion filtering is described in terms of time, therefore the scale unit are mapped to time units as given below:

$$t_i = \frac{1}{2}\sigma_i^2, i = \{0,1,...N\} \tag{2.37}$$

After the formulation of the scale space, it is required to detect the key points having certain characteristics such as robustness against image transformations, scale and position independence. A detailed overview of KAZE featuers can be seen in [Alcantarilla *et al.*, 2012].