CHAPTER 4

HIERARCHICAL PHASE BASED MATCHING

4.1 MOTIVATION

In this chapter a hierarchical methodology of phase based image matching technique is introduced that describes the extraction of phase components from the subregions extracted from the iris images and also the matching process using the phase component. Since it is a phase based matching, the phase components of the subregion of irises are extracted using 2D DFT. The extracted subregions are compared with the database using BLPOC [Band Limited Phase Only Correlation]. And depending upon the peak the comparison exhibits, the matching score is calculated and thresholded for the discrimination of matched and non matched component. Each subregions are matched in each level of the hierarchical matching, which reduces the “to be matched” set of database drastically in each level in an exponential order thereby reduces the time taken for matching.

4.2 2-D DISCRETE FOURIER TRANSFORM

The Fourier transform, named for Joseph Fourier[^81], is a mathematical transform with many applications in physics and engineering. Very commonly, it expresses a mathematical function of time as a function of frequency, known as its frequency spectrum. The Fourier integral theorem details this relationship. For instance, the transform of a musical chord made up of pure notes (without overtones) expressed as amplitude as a function of time, is a mathematical representation of the amplitudes and phases of the individual notes that make it up. The function of time is often called the time
domain representation, and the frequency spectrum the frequency domain representation. The inverse Fourier transform expresses a frequency domain function in the time domain. Each value of the function is usually expressed as a complex number (called complex amplitude) that can be interpreted as a magnitude and a phase component. The term "Fourier transform" refers to both the transform operation and to the complex-valued function it produces.

In the case of a periodic function, such as a continuous, but not necessarily sinusoidal, musical tone, the Fourier transform can be simplified to the calculation of a discrete set of complex amplitudes, called Fourier series coefficients. Also, when a time-domain function is sampled to facilitate storage or computer-processing, it is still possible to recreate a version of the original Fourier transform according to the Poisson summation formula, also known as discrete-time Fourier transform.

The Fourier transform allows showing the different parts of a continuous signal. These are often used to calculate the frequency spectrum of a signal that changes over time. Such signals are found in many applications, like recognizing speech or handwriting, but also in cryptography or oceanography. A Fourier transform really just shows, what frequencies are in a signal. For instance, a sound wave, that has the musical notes A and B and C in it, the Fourier transform of that signal will show the frequency on the x-axis, with peaks in the graph at the frequencies of the notes A, B, and C.

Many signals can be created by adding together many cosines and sines of different amplitudes and frequencies. The Fourier transform is just the amplitude and phases of these cosines and sines plotted against their respective frequency. Fourier transforms are important because many signals make more sense when you look at what frequencies are in them,
rather than when looked at the signal with time on the x-axis. Everyone discusses the Fourier transform when discussing signal processing. Why is it so important to image processing and what does it tell us about the signal? Does it only apply to image processing or does it apply to other signals as well?

This is quite a broad question and it indeed is quite hard to pinpoint why exactly Fourier transforms are important in image processing. The simplest, hand waving answer one can provide is that it is an extremely powerful mathematical tool that allows you to view your signals in a different domain, inside which several difficult problems become very simple to analyze. It provides one-to-one transform of signals from/to a time-domain representation \( f(t) \) to/from a frequency domain representation \( F(\delta) \). It allows a frequency content (spectral) analysis of a signal. FT is suitable for periodic signals. If the signal is not periodic then the Windowed FT or the linear integral transformation with time (spatially in 2D) localized basis function, e.g., wavelets, Gabor filters can be used.

The **Discrete Fourier Transform (DFT)**\(^{[82]}\) is a specific kind of discrete transform, used in Fourier analysis. It transforms one function into another, which is called the frequency domain representation, or simply the DFT, of the original function (which is often a function in the time domain). The DFT requires an input function that is discrete. Such inputs are often created by sampling a continuous function, such as a person's voice. The discrete input function must also have a limited (finite) duration, such as one period of a periodic sequence or a windowed segment of a longer sequence. The inverse DFT cannot reproduce the entire time domain, unless the input happens to be periodic. Therefore it is
often said that the DFT is a transform for Fourier analysis of finite-domain discrete-time functions.

The input to the DFT is a finite sequence of real or complex numbers making the DFT ideal for processing information stored in computers. In particular, the DFT is widely employed in signal processing and related fields to analyze the frequencies contained in a sampled signal, to solve partial differential equations, and to perform other operations such as convolutions or multiplying large integers. A key enabling factor for these applications is the fact that the DFT can be computed efficiently in practice using a fast Fourier transform (FFT) algorithm. The sequence of N complex numbers $x_0, \ldots, x_{N-1}$ is transformed into another sequence of N complex numbers according to DFT Formula: 

$$\sum_{n=0}^{N-1} x_n$$

The transform is sometimes denoted by the symbol $F$, as in $F\{x\}$ or $F(x)$ or $Fx$. Let $f(x)$ be an input signal (a sequence), $x = 0, \ldots, N - 1$. Let $F(k)$ be a Frequency spectrum (the result of the discrete Fourier transformation) of a signal $f(k)$. Then the Discrete Fourier transformation is given by

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-2\pi i kn / N}$$

Some Properties of DFT:

1. **Completeness:**

   The discrete Fourier transform is an invertible, linear transformation where $F: \mathbb{C}^N \rightarrow \mathbb{C}^N$ with $\mathbb{C}$ denoting the set of complex numbers. In other words, for any $N > 0$,
an N-dimensional complex vector has a DFT and an IDFT which are in turn N-dimensional complex vectors.

2. Orthogonality:

The vectors, \( u_k = \left[ e^{2\pi i k n} \mid n = 0, 1, \ldots, N - 1 \right]^T \), form an orthogonal basis over the set of N-dimensional complex vectors:

\[
\begin{align*}
    u_k^T u_{k'}^* &= \sum_{n=0}^{N-1} \left( e^{2\pi i k n} \right) \left( e^{2\pi i (-k') n} \right) = \sum_{n=0}^{N-1} e^{2\pi i (k-k') n} = N \delta_{kk'}
\end{align*}
\]

where \( \delta_{kk'} \) is the Kronecker delta. (This orthogonality condition can be used to derive the formula for the IDFT from the definition of the DFT, and is equivalent to the unitarity property below.

3. The Plancherel theorem and Parseval's theorem:

If \( X_k \) and \( Y_k \) are the DFTs of \( x_n \) and \( y_n \) respectively then the Plancherel theorem states:

\[
\sum_{n=0}^{N-1} x_n y_n^* = \frac{1}{N} \sum_{k=0}^{N-1} X_k Y_k^*
\]

where the star denotes complex conjugation. Parseval's theorem is a special case of the Plancherel theorem and states that:

\[
\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2.
\]

These theorems are also equivalent to the unitary condition explained below.

4. Periodicity:
If the expression that defines the DFT is evaluated for all integers \( k \) instead of just for \( k=1, \ldots, N-1 \), then the resulting infinite sequence is a periodic extension of the DFT, periodic with period \( N \).

The periodicity can be shown directly from the definition:

\[
X_{k+N} \overset{\text{def}}{=} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} (k+N)n} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} e^{-2\pi i n} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} = X_k.
\]

Similarly, it can be shown that the 1 DFT formula leads to a periodic extension.

5. The shift theorem:

Multiplying \( x_n \) by a linear phase \( e^{\frac{2\pi imn}{N}} \) for some integer \( m \) corresponds to a circular shift of the output \( X_k \): \( X_k \) is replaced by \( X_{k-m} \), where the subscript is interpreted modulo \( N \) (i.e., periodically). Similarly, a circular shift of the input \( x_n \) corresponds to multiplying the output \( X_k \) by a linear phase. Mathematically, if \( \{x_n\} \) represents the vector \( x \) then

\[
\text{If } F(\{x_n\}) k = X_k
\]

then \( F\left(\{x_n e^{\frac{2\pi imn}{N}}\}\right)_k = X_{k-m} \)

and \( F(\{x_{n-m}\})_k = X_k e^{-\frac{2\pi ikm}{N}} \)

6. Circular convolution theorem and cross-correlation theorem:

The convolution theorem for the discrete-time Fourier transform indicates that a convolution of two infinite sequences can be obtained as the inverse transform of the product of the individual transforms. An important simplification occurs when the sequences are of finite length, \( N \). In terms of the DFT and inverse DFT, it can be written as follows:
\[
\mathbf{F}^{-1}\{\mathbf{X,Y}\}_n = \sum_{l=0}^{N-1} x_l \cdot (Y_N)_{n-l} \overset{\text{def}}{=} (x \ast y)_n
\]

which is the convolution of the \(\mathbf{X}\) sequence with a \(\mathbf{Y}\) sequence extended by periodic summation:

\[
(Y_N)_n \overset{\text{def}}{=} \sum_{p=-\infty}^{\infty} y_{(n-N)} = y_{n(\text{mod}N)}
\]

Similarly, the cross-correlation of \(\mathbf{X}\) and \(Y\) is given by:

\[
\mathbf{F}^{-1}\{\mathbf{X^*,Y}\}_n = \sum_{l=0}^{N-1} x^*_l \cdot (Y_N)_{n+l} \overset{\text{def}}{=} (x^* \cdot y_N)_n
\]

A direct evaluation of either summation (above) requires \(O(N^2)\) operations for an output sequence of length \(N\). An indirect method, using transforms, can take advantage of the \(O(N \log N)\) efficiency of the fast Fourier transform (FFT) to achieve much better performance. Furthermore, convolutions can be used to efficiently compute DFTs via Rader's FFT algorithm and Bluestein's FFT algorithm. Methods have also been developed to use circular convolution as part of an efficient process that achieves normal (non-circular) convolution with an \(\mathbf{x}\) or \(\mathbf{y}\) sequence potentially much longer than the practical transform size (\(N\)). Two such methods are called overlap-save and overlap-add.

7. The unitary DFT:

Another way of looking at the DFT is to note that in the above discussion, the DFT can be expressed as a Vandermonde matrix:
Where, 

$$\omega_N = e^{-2\pi i/N}$$

is a primitive Nth root of unity. The inverse transform is then given by the inverse of the above matrix:

$$F^{-1} = \frac{1}{N} F^*$$

8. Expressing the inverse DFT in terms of the DFT:

A useful property of the DFT is that the inverse DFT can be easily expressed in terms of the (forward) DFT, via several well-known "tricks". (For example, in computations, it is often convenient to only implement a fast Fourier transform corresponding to one transform direction and then to get the other transform direction from the first.) The inverse DFT can be computed by reversing the inputs:

$$F^{-1}(\{x_n\}) = \frac{1}{N} F(\{x_{N-n}\})$$

9. Eigen Values and Eigen Vectors:

The eigen values of the DFT matrix are simple and well-known, whereas the eigen vectors are complicated, not unique, and are the subject of ongoing research.
$\sum_{m,n} = \frac{1}{\sqrt{N}} \omega_N^{(m-1)(n-1)} = \frac{1}{\sqrt{N}} e^{2\pi i (m - 1)(n - 1)}$

10. Uncertainty principle:

If the random variable $X_k$ is constrained by:

$$\sum_{n=0}^{N-1} |X_n|^2 = 1$$

then $P_n = |X_n|^2$ may be considered to represent a discrete probability mass function of $n$, with an associated probability mass function constructed from the transformed variable:

$$Q_m = N |x_m|^2$$

2Dimensional DFT

If we have a real signal $x(k)$, $k = 1, \ldots, N$, where $N$ is a power of 2, the Discrete Fourier Transform produces an array $X(1 : N)$ of $N$ complex numbers. Complex numbers $a + ib$ have a real part ‘a’ and an imaginary part ‘b’. Instead of talking about one dimensional signals that represent changes in amplitude in time, here the dealing is with two dimensional signals which represent intensity variations in space. These signals come in the form of images. The digital images have a finite width and height in pixels, which is assumed to have a real number value. Because our signals are discrete, we will need an analog of the one dimensional DFT for two dimensional signals. This analog is the following pair of transforms:
F(u, v) = \frac{1}{MN} + \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \exp \left[-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right]

Where u=0,1,…M-1 & v=0,1,…..N-1

f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi \left(\frac{ux}{M} + \frac{vy}{N}\right)}

Thus an MxN image has an MxN set of (complex) fourier coefficients. To implement this transform, the coefficients of the transform is computed quickly. The two dimensional DFT is separable into two one dimensional DFTs which can be implemented with an FFT algorithm.

F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j2\pi ux/N} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi vy/N}

The outcome of the Fourier transform F(u, v) is a function of complex variables.

Matrix Form of 2D DFT

When reconsidering the 2D DFT:

X[k, l] = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \left(\sum_{m=0}^{M-1} x[m, n] e^{-j2\pi \frac{mk}{M}}\right) e^{-2\pi \frac{nl}{N}} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X'[k, n] e^{-j2\pi \frac{nl}{N}}

(k = 0,1, …., M − 1)

where 0<=m,k<=N-1,0<=n,l<=N-1 and

X'[k,n] \triangleq \frac{1}{\sqrt{M}} \sum_{m=0}^{N-1} x[m, n] e^{-j2\pi \frac{mk}{M}} \quad (n=0,1,….N-1)
As the summation above is with respect to the row index ‘m’ and the column index ‘n’ which can be treated as a fixed parameter, this expression can be considered as a one-dimensional Fourier transform of the nth column of \([x](x=0,1,\ldots,N-1)\). The 2D DFT maps an \(M \times N\) scalar picture \(f\) into a complex-valued Fourier transform \(F\). This is also called a mapping from the spatial domain (of pictures) into the frequency domain (of Fourier transforms). The benefit is that changes in the frequency domain can be directed on particular components in the picture, defined by combinations of sines and cosines. After those changes, called filter operations in the frequency domain, an inverse 2D DFT maps the modified Fourier transform back into a modified picture. The whole process is called Fourier filtering, and allows, for example, contrast enhancement, noise removal, or smoothing of the picture. 1D Fourier filtering is commonly used in signal theory (e.g., for voice processing or recognition), and 2D Fourier filtering of pictures follows the same principles (just in 2D instead of 1D).

**Properties of 2D DFT:**

As with the one dimensional DFT, there are many properties of the transformation that give insight into the content of the frequency domain representation of a signal and allow us to manipulate signals in one domain or the other.

1) **Shift Property:**

As in one dimension, there is a simple relationship that can be derived for shifting an image in one domain or the other. Since both the space and frequency domains are considered periodic for the purposes of the transforms, shifting means rotating around the boundaries. The equations describing this are:
\[
F(x,y)e^{\frac{j2\pi(u_0x+y_0y)}{N}} \Leftrightarrow F(u-u_0, v-v_0)
\]

\[
F(x-x_0, y-y_0) \Leftrightarrow F(u, v)e^{-j2\pi(u_0x+y_0y)/N}
\]

2) **Scale Property:**

Just as in one dimension, shrinking in one domain causes expansion in the other for the 2D DFT. This means that as an object grows in an image, the corresponding features in the frequency domain will expand. The equation governing this is:

\[
F(ax,by) \rightarrow \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)
\]

3) **Rotation Property:**

This is a property of the 2D DFT that has no analog in one dimension. Because of the separability of the transform equations, the content in the frequency domain is positioned based on the spatial location of the content in the space domain. This means that rotating the spatial domain contents rotates the frequency domain contents. This can be formally described by the following relationship:

\[
f(x,y) \leftrightarrow F(u,v)
\]

\[
f(x',y') \leftrightarrow F(u',v')
\]

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} = \begin{pmatrix}
  \cos\theta & -\sin\theta \\
  \sin\theta & \cos\theta
\end{pmatrix} \begin{pmatrix}
  x \\
  y
\end{pmatrix}
\]

\[
\begin{pmatrix}
  u' \\
  v'
\end{pmatrix} = \begin{pmatrix}
  \cos\theta & -\sin\theta \\
  \sin\theta & \cos\theta
\end{pmatrix} \begin{pmatrix}
  u \\
  v
\end{pmatrix}
\]
Multidimensional DFT

The ordinary DFT transforms a one-dimensional sequence or array \( x_n \) that is a function of exactly one discrete variable \( n \). The multidimensional DFT of a multidimensional array \( x_{n_1, n_2, \ldots, n_d} \) that is a function of \( d \) discrete variables \( n_l = 0, 1, 2, \ldots, N_l - 1 \) for \( l \) in \( 1, 2, \ldots, d \) is defined by:

\[
X_{k_1, k_2, \ldots, k_d} = \sum_{n_1=0}^{N_1-1} \left( \omega_{N_1}^{k_1 n_1} \sum_{n_2=0}^{N_2-1} \left( \omega_{N_2}^{k_2 n_2} \cdots \sum_{n_d=0}^{N_d-1} \omega_{N_d}^{k_d n_d} \cdot x_{n_1, n_2, \ldots, n_d} \right) \right)
\]

where,

\[
w_{N_l} = \exp(-2\pi i / N_l)
\]
as above and the \( d \) output indices run from \( k_l = 0, 1, \ldots, N_l - 1 \)

2D DFT in Image Processing

The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression. The basis vectors for coordinate system tell point as linear combination of orthogonal basis vectors: \( x = a_1 v_1 + \ldots + a_n v_n \). The standard basis for images is the set of unit vectors corresponding to each pixel. The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, i.e. the image in the spatial and Fourier domain are of the same size. For a square image of size \( N \times N \), the two-dimensional DFT is given by:
\[ F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi \frac{ki}{N^2} + \frac{lj}{N^2}} \]

where \( f(a,b) \) is the image in the spatial domain and the exponential term is the basis function corresponding to each point \( F(k,l) \) in the Fourier space. The equation can be interpreted as: the value of each point \( F(k,l) \) is obtained by multiplying the spatial image with the corresponding base function and summing the result. The basis functions are sine and cosine waves with increasing frequencies, i.e. \( F(0,0) \) represents the DC-component of the image which corresponds to the average brightness and \( F(N-1,N-1) \) represents the highest frequency. In a similar way, the Fourier image can be re-transformed to the spatial domain.

The inverse Fourier transform is given by:

\[ F(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{2\pi \frac{ika}{N} \frac{lb}{N}} \]

Note the \( 1/N^2 \) normalization term in the inverse transformation. This normalization is sometimes applied to the forward transform instead of the inverse transform, but it should not be used for both. To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is separable, it can be written as

\[ F(k,l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k,b) e^{-\frac{2\pi lb}{N}} \]

Where,

\[ P(k,b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a,b) e^{-\frac{2\pi ka}{N}} \]
Using these two formulas, the spatial domain image is first transformed into an intermediate image using $N$ one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using None-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of $2N$ one-dimensional transforms decreases the number of required computations.

The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the real and imaginary part or with magnitude and phase. In image processing, only the magnitude of the Fourier Transform is displayed often, as it contains most of the information of the geometric structure of the spatial domain image. However, to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, it is made sure to preserve both magnitude and phase of the Fourier image. The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values. The DFT has seen wide usage across a large number of fields. All applications of the DFT depend crucially on the availability of a fast algorithm to compute discrete Fourier transforms and their inverses, a fast Fourier transform.

Subregion in Space Domain                                      Subregion in Frequency Domain

Fig 4.1: 2D DFT representation of an Iris Subregion
1) Spectral analysis

When the DFT is used for spectral analysis, the \( \{x_n\} \) sequence usually represents a finite set of uniformly spaced time-samples of some signal \( x(t) \), where \( t \) represents time. The conversion from continuous time to samples (discrete-time) changes the underlying Fourier transform of \( x(t) \) into a discrete-time Fourier transform (DTFT), which generally entails a type of distortion called aliasing. Choice of an appropriate sample-rate (see Nyquist rate) is the key to minimizing that distortion. Similarly, the conversion from a very long (or infinite) sequence to a manageable size entails a type of distortion called leakage, which is manifested as a loss of detail (aka resolution) in the DTFT. Choice of an appropriate sub-sequence length (see Coherent sampling) is the primary key to minimizing that effect.

2) Data compression

The field of digital signal processing relies heavily on operations in the frequency domain (i.e. on the Fourier transform). For example, several lossy image and sound compression methods employ the discrete Fourier transform: the signal is cut into short segments, each is transformed, and then the Fourier coefficients of high frequencies, which are assumed to be unnoticeable, are discarded. The decompressor computes the inverse transform based on this reduced number of Fourier coefficients. (Compression applications often use a specialized form of the DFT, the discrete cosine transform or sometimes the modified discrete cosine transform.)

3) Partial differential equations

Discrete Fourier transforms are often used to solve partial differential equations, where again the DFT is used as an approximation for the Fourier series (which is recovered in the limit of
infinite $N$). The advantage of this approach is that it expands the signal in complex exponentials $e^{inx}$, which are eigenfunctions of differentiation: $d/dx e^{inx} = in e^{inx}$. Thus, in the Fourier representation, differentiation is simple—we just multiply by $in$. (Note, however, that the choice of $n$ is not unique due to aliasing; for the method to be convergent, a choice similar to that in the trigonometric interpolation section above should be used.) A linear differential equation with constant coefficients is transformed into an easily solvable algebraic equation. One then uses the inverse DFT to transform the result back into the ordinary spatial representation. Such an approach is called a spectral method.

4) Polynomial multiplication

Suppose we wish to compute the polynomial product $c(x) = a(x) \cdot b(x)$. The ordinary product expression for the coefficients of $c$ involves a linear (acyclic) convolution, where indices do not "wrap around." This can be rewritten as a cyclic convolution by taking the coefficient vectors for $a(x)$ and $b(x)$ with constant term first, then appending zeros so that the resultant coefficient vectors $c = a \ast b$ have dimension $d > \deg(a(x)) + \deg(b(x))$.

5) Multiplication of large integers

The fastest known algorithms for the multiplication of very large integers use the polynomial multiplication method outlined above. Integers can be treated as the value of a polynomial evaluated specifically at the number base, with the coefficients of the polynomial corresponding to the digits in that base. After polynomial multiplication, a relatively low-complexity carry-propagation step completes the multiplication.
4.3 PHASE COMPONENT IN 2D DFT

The input waveform will in general contain frequencies of various phase angles – they need not be sine waves. To detect a frequency component of some arbitrary phase, it is necessary to use both inphase and quadrature search frequencies. Then the magnitude of frequency component can be obtained applying the hypotenuse formula to the inphase and quadrature components:

$$\text{Magnitude} = \sqrt{\text{inphase}^2 + \text{quadrature}^2}$$

This is the most used result of a spectrum analysis. In some situations, the phase of the spectral component is also useful:

$$\text{Phase} = \tan^{-1}\left(\frac{\text{quadrature}}{\text{inphase}}\right)$$

There is one further wrinkle to the DFT, and that is the matter of negative frequencies. It is normally thought of a sine wave as the result of a vector that rotates in a clockwise direction. However, a given sine wave can also be visualized as the sum of two vectors, one rotating clockwise, the other counterclockwise. The vectors are equal in length, each is half of the total. The clockwise rotating vector may be regarded as a positive frequency. The counterclockwise rotating vector is a negative frequency.

The outcome of the Fourier transform $F(u, v)$ is a function of complex variables which is represented as (Complex) spectrum $F(u, v) = R(u, v) + i I(u, v)$. The Phase spectrum is given as:

$$\varphi(u, v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right].$$
Here, the Magnitude determines the contribution of each component and the Phase determines which components are present. Instead of representing the complex numbers as real and imaginary parts we can represent it as Magnitude and Phase where they are defined as:

\[\text{Magnitude}(f) = \sqrt{\text{Re}^2 + \text{Im}^2}\]
\[\text{Phase}(f) = \arctan\left(\frac{\text{Im}}{\text{Re}}\right)\]

Here, the Magnitude is telling how much of a certain frequency component is in the image. Phase is telling where that certain frequency lies in the image.

4.4 PHASE AS A FEATURE CODE FOR MATCHING

4.4.1 Phase Only Correlation [POC]

Phase only correlation (POC) is a kind of limited correlation and can estimate translational displacements, rotation and scaling with subpixel accuracy between two images. The study of its application in areas such as biometrics, image motion analysis and video retrieval has advanced. If POC is to be applied to coded images, they should first be decoded. Moreover, because a phase term is a complex number, the use of the POC requires more physical space and time. The definition of a phase only correlation is given as:

Consider two \(N1 \times N2\) images, \(f(n1,n2)\) and \(g(n1,n2)\), where we assume that the index ranges are \(n1 = -M1\ldots M1\) \((M1>0)\) and \(n2 = -M2\ldots M2\) \((M2>0)\) for mathematical simplicity, and hence \(N1=2M1+1\) and \(N2=2M2+1\). Let \(F(k1,k2)\) and \(G(k1,k2)\) denote the 2D Discrete Fourier Transforms (2D DFTs) of the two images. \(F(k1,k2)\) and \(G(k1,k2)\) are given by
\[
F(k1,k2) = \sum_{n1,n2} f(n1,n2) W_{N1}^{k1n1} W_{N2}^{k2n2}
= A_F(k1,k2) e^{j\theta_F(k1,k2)}
\]
\[
G(k1,k2) = \sum_{n1,n2} g(n1,n2) W_{N1}^{k1n1} W_{N2}^{k2n2}
= A_G(k1,k2) e^{j\theta_G(k1,k2)}
\]

Where \( k1 = -M1 \ldots M1, \) \( k2 = -M2 \ldots M2, \) \( W_{N1} = e^{j2\pi M1}, \) \( W_{N2} = e^{j2\pi M2}, \) and the operator \( \sum n1,n2 \) denotes

\[ \sum_{n1}^{M1} = -M1 \text{ and } \sum_{n2}^{M2} = -M2. \]

\( A_F(k1,k2) \) and \( A_G(k1,k2) \) are amplitude components, and \( e^{j\theta_F(k1,k2)} \) and \( e^{j\theta_G(k1,k2)} \) are phase components. The cross spectrum \( R_{FG} (k1,k2) \) between \( F(k1,k2) \) and \( G(k1,k2) \) is given by

\[
R_{FG} (k1,k2) = F(k1,k2)G(k1,k2)
= A_F(k1,k2) A_G(k1,k2) e^{j\theta(k1,k2)}
\]

Where \( G(k1,k2) \) denotes the complex conjugate of \( G(k1,k2) \) and \( \theta(k1,k2) \) denotes the phase difference \( \theta_F(k1,k2) - \theta_G(k1,k2). \)

Phase-Only Correlation (POC) has been widely used for image registration tasks, and recently it has been adopted in some biometric systems as a similarity measure. POC based method relies on the translation property of the Fourier transform. Let \( f \) and \( g \) be the two images that differ only by a displacement \( (x_0, y_0) \) i.e. \( g(x,y)=f(x-x_0,y-y_0). \)

Their corresponding Fourier transforms \( G(u,v) \) and \( F(u,v) \) will be related by

\[
G(u,v)=e^{-j2\pi(ux_0+vy_0)}F(u,v)
\]

The cross-phase spectrum \( R_{GF} (u,v) \) between \( G(u,v) \) and \( F(u,v) \) is given by
where $F^*$ is the complex conjugate of $F$. By taking inverse Fourier transform of RGF back to
the time domain, we will have a Dirac impulse centered on $(x_0, y_0)$. In practice, we should
consider the finite discrete representations. Consider two $M \times N$ images, $f(m,n)$ and $g(m,n)$,
where the index ranges are $m=-M_0, \ldots, M_0$ ($M_0 > 0$) and $n=-N_0, \ldots, N_0$ ($N_0 > 0$), and $M =
2M_0+1$ and $N = 2N_0+1$. Denote by $F(u,v)$ and $G(u,v)$ the 2D DFTs of the two images and
they are given by

$$
F(u,v) = \sum_{m=-M_0}^{M_0} \sum_{n=-N_0}^{N_0} f(m,n) e^{-j2\pi \left( \frac{mu}{M} + \frac{nv}{N} \right)} = A_F(u,v) e^{j\phi_F(u,v)}
$$

$$
G(u,v) = \sum_{m=-M_0}^{M_0} \sum_{n=-N_0}^{N_0} g(m,n) e^{-j2\pi \left( \frac{mu}{M} + \frac{nv}{N} \right)} = A_G(u,v) e^{j\phi_G(u,v)}
$$

where $u=-M_0, \ldots, M_0$, $v=-N_0, \ldots, N_0$, $A_F(u,v)$ and $A_G(u,v)$ are amplitude components, and
$\phi_F(u,v)$ and $\phi_G(u,v)$ are phase components. Then, the cross-phase spectrum $RGF(u,v)$
between $G(u,v)$ and $F(u,v)$ is given by

$$
RGF(u,v) = \frac{G(u,v)F^*(u,v)}{|G(u,v)F^*(u,v)|} = e^{j(\phi_G(u,v) - \phi_F(u,v))}
$$

The POC function $pgf(m,n)$ is the 2D Inverse DFT (IDFT) of $RGF(u,v)$:

$$
p_{gf}(m,n) = \frac{1}{MN} \sum_{u=-M_0}^{M_0} \sum_{v=-N_0}^{N_0} RGF(u,v) e^{j2\pi \left( \frac{mu}{M} + \frac{nv}{N} \right)}
$$
Fig 4.2: Phase Only correlation representing the sharp peak for similar images
If the two images f and g are similar, their POC function will give a distinct sharp peak. On the contrary, if they are not similar, the peak value will drop significantly. Thus, the height of the peak value can be used as a similarity measure, and the location of the peak shows the translational displacement between the two images.

The most common stereo correspondence techniques employ Sum of Absolute Differences (SAD) or Sum of Squared Differences (SSD), where corresponding points between stereo images can be obtained by minimizing SAD or SSD in area-based block matching. Although SAD and SSD exhibit low computational cost, a major drawback is their low accuracy. Recently, sub-pixel block matching techniques using SAD and SSD have been investigated [4], but the obtained accuracy is not sufficient in some applications. On the other hand, image matching methods using 2D Phase-Only Correlation (POC) exhibit much better matching performance than the methods using SAD and SSD in general.

Steps done for POC:

1) First of all, different frequency information from image using DFT technique is got. Then, Phase correlation of source and templates frequencies (got from source and template images, both are n-dimensional) is done.

2) After phase only correlation, do inverse Fourier and find the peaks in the real part. That peaks shows the matching positions.

3) After this phase correlation using the below equation is applied.

\[(a+ib)(p+iq)^* / (| (a+ib)(p+iq)^*|)\]

where \((a+ib)\) represents the complex number got from source DFT (so for NxN dimensional image, there will be N*N complex numbers, represented as 2D array
(p+iq) represents the complex number got from template DFT and (p+iq)* is conjugate operation.

4) The above equation gives high values for signals where peaks and bottoms correctly aligns each other.

5) The inverse DFT on the phase only correlated data is done, and some threshold on the real part in order to get the peaks which indicates the matched positions is applied.

4.4.2 Band-Limited Phase-Only Correlation [BLPOC]

In the POC-based image matching method, all the frequency components are involved. However, high frequency tends to emphasize detail information and can be prone to noise. To eliminate meaningless high frequency components, K. Ito et al.\cite{83} proposed the Band-Limited Phase-Only Correlation (BLPOC).

The BLPOC limits the range of the spectrum of the given image. Assume that the ranges of the inherent frequency band of texture are given by \( u=-U_0, \ldots, U_0 \) and \( v=-V_0, \ldots, V_0 \), where \( 0<=U_0<=M_0, 0<=V_0<=N_0 \). Thus, the effective size of spectrum is given by \( L_1=2U_0+1 \) and \( L_2=2V_0+1 \). The BLPOC function is defined as

\[
p_{gf}^{U_0V_0}(m,n)=\frac{1}{L_1L_2}\sum_{u=-U_0}^{U_0}\sum_{v=-V_0}^{V_0}R_{gf}(u,v)e^{j2\pi\frac{mu}{L_1}\frac{nv}{L_2}}\]

where \( m=-U_0, \ldots, U_0 \) and \( n=-V_0, \ldots, V_0 \). When two images are similar, their BLPOC function gives a distinct sharp peak. Also, the translational displacement between the two images can be estimated by the correlation peak position. Experiments indicate that the BLPOC function provides a much higher discrimination capability than the original POC function. It adaptively changes the size of the 2D IDFT. The figure below represents how the BLPOC function shows higher discrimination capability than the original POC function.
Subregions in Spacial Domain

Frequency Domain (Effective Frequency Band)

Max = 0.452

Fig 4.3: Representation of BLPOC
4.5 HIERARCHICAL IMPLEMENTATION OF PHASE BASED MATCHING

In this hierarchical matching, consider an aligned iris image \( f(n_1, n_2) \). The BLPOC function is calculated for the latent image \( f(n_1, n_2) \) and let the phase component get be \( \theta_1 \) and the matching score is evaluated with minimum two database images \( g(n_1, n_2) \) and \( h(n_1, n_2) \) hierarchically. If the phase component \( \theta_1 \) matches any database image either \( g(n_1, n_2) \) or \( h(n_1, n_2) \), then it will return the matching score value.

Matching can be seen as traversing the tree structure of templates. The matching process starts at the root, the interest locations lie initially on a uniform grid over relevant regions in the image. The tree can be traversed in breadth first or depth first fashion.

In the proposed method, the top-to-bottom approach is used. The top-down sequence follows the nodes from the root to the leaf. Its principle functionality is indicated in the following code fragment:

01. Input_root_image(i)
02. for each L in leaf(i)
03.   if \( f(L) = i \)
04.     marknode_select(L)
05.     return marknode_select(L) for matching score calculation
06.   if genuine matching
07.     return matching score
08.   back to leaf
09.   else
10.   top_down_evaluate(s)
The function Input_root_image gets the actual node as a parameter, which initially is the root of the search graph (line 1). Each leaf L of the current node is addressed in a loop (line 2) and for each L the formula f(L) is evaluated. If f(L) is true, the current leaf is marked as relevant (line 4). If f(L) fails, the recursion is continued until there is a node which fulfills the formula or until no subsequent node is left (line 6). When applying this code fragment on the iris recognition, the Input_root_image get the latent iris image which is the root of the search graph. The loop foreach is used to go through the leaves, and then if a leaf node matches with the root node, it will be marked as relevant image and return the image for matching score calculation. Else if it fails to match in the matching score calculation it will return and the recursion will be continued until it matches.

So in the place of matching with a single database image, two database images can be matched hierarchically with the input image. So by expanding the hierarchical images the speed will increase.

The flow chart for the proposed hierarchical matching is shown in the Fig. X. In the proposed method a hierarchical matching in which, the phase component values are calculated for the subregions alone for the iris alone is introduced. So that the image will not be affected by any reflection and also the phase component calculating value and time will be reduced.

In the proposed algorithm Hierarchical method of phase based matching is implemented as follows: Let the input image be f(n1, n2) and the boundaries of the images are detected and while detecting it will return the center coordinate value and the radius value of the iris and the pupil. Then using the center coordinate values and the radius values along
with the proposed HPM algorithm, the subregions are calculated, in a manner that the
subregion boundaries should not exceed the iris boundary.

**ALGORITHM:**

(Calculate the phase component for S1, S2, S3, S4, S5). Let the phase component for the
subregions be $\theta_{S1}(x, y)$, $\theta_{S2}(x, y)$, $\theta_{S3}(x, y)$, $\theta_{S4}(x, y)$, $\theta_{S5}(x, y)$ for S1, S2, S3, S4, S5
respectively.)

(i) Calculate the BLPOC for the subregion S1 of input image $f(n1, n2)$ with the
subregion S1 of the database images M.

(ii) Calculate the matching score for those subregion
    If BLPOC value of S1 for $f(n1, n2) = S1$ of some N images of M.
    Then
    Get the matched N images; Where N<M;

(iii) Calculate the BLPOC for the subregion S2 of $f(n1, n2)$ with the S2 of the images in N
    If BLPOC value of S2 for $f(n1, n2) = S2$ of some W images of N.
    Then
    Get the matched W images; Where X<W;

(iv) Calculate the BLPOC for the subregion S3 of $f(n1, n2)$ with the S3 of the images in W.
    If BLPOC value of S3 for $f(n1, n2) = S3$ of some X images of W.
    Then
    Get the matched X images; Where X<W;

(v) Calculate the POC for the subregion S4 of $f(n1, n2)$ with the S4 of the images in Y
    If POC value of S4 for $f(n1, n2) = S4$ of some Y images of X.
    Then
    Get the matched Y images; Where Y<X;

(vi) Calculate the POC for the subregion S5 of $f(n1, n2)$ with the S5 of the images in Y
    If POC value of S5 for $f(n1, n2) = S5$ of some Z images of Y.
    Then
    Get the matched Z image(s). Where Z < Y;

Fig 4.4 Algorithm for Hierarchical Phase Based Matching

118
Band Limited Phase Only Correlation (BLPOC) of Sub Region 1 with 'M'

'N' matched output; N<M, where M is the database size

BLPOC of Sub Region 2 with 'N'

'W' matched output; W<N

BLPOC of Sub Region 3 with 'W'

'X' matched output; X<W

BLPOC of Sub Region 4 with 'X'

'Y' matched output; Y<X

BLPOC of Sub Region 5 with 'Y' that outputs 'Z' match

LEVEL-1
LEVEL-2
LEVEL-3
LEVEL-4
LEVEL-5

Fig 4.5: Hierarchical Phase Based Matching-Flow Chart
The hierarchical matching is implemented in the matching stage. Here instead of calculating the phase component for the entire image, we are calculating the phase component for the five subregions (S1, S2, S3, S4, and S5) of the iris.

In the first level of hierarchical matching, the subregion S1 is taken from the input image \( f(n_1, n_2) \) and the BLPOC value for the S1 of \( f(n_1, n_2) \) with the S1 of the M database images is calculated. The matched shortlisted images be N, where N<M.

In the second level, the shortlisted images N are subjected to matching. The BLPOC values for the subregion S2 of \( f(n_1, n_2) \) is calculated with the S2 of the shortlisted images. This outputs W number of matched images, where W<N.

The third level consists of matching the W shortlisted images. The BLPOC values for the subregion S3 of \( f(n_1, n_2) \) is calculated with the S3 of the shortlisted images. This outputs X number of matched images, where X<W.

In the forth level, the shortlisted images X are subjected to matching. The BLPOC values for the subregion S4 of \( f(n_1, n_2) \) is calculated with the S4 of the shortlisted images. This outputs Y number of matched images, where Y<X.

Finally in the fifth level, by applying the BLPOC function for the subregion S5 of \( f(n_1, n_2) \) with the S5 of Y, Z number of another shortlisted image is driven, where Z<Y.

4.6 SUMMARY

By this hierarchical method of matching, the threshold value of calculating the BLPOC is reduced. Compared with the threshold value for calculating the BLPOC for a normalized image of pixel value (256x128), the proposed method can produce better performance. The results obtained and the performances are analyzed in the further chapters.