CHAPTER 3

FACE COMPARISION USING PARTICLE SWARM OPTIMIZATION

While compressing images, the foremost goal is to lessen the insignificance and redundancy of the image data so as to ensure efficient pile and transfer of data. At times, there are chances for losing some predominant image data. In other words, preserving image eminence at a given bit-rate or density rate is the main objective of image density. Compression methods with tolerable degradation of quality are required.

With the growth of technology, it is tedious to deal with vast amount of information. Image compression accelerates in diminishing the size of a graphics file without humiliating the eminence of the image to an intolerable level. This reduction permits more images to be stored in a given remembrance. It also decreases the transmission and downloading time.

JPEG and JPEG 2000 are two important procedures used for image compression. JPEG image compression procedure uses Discrete Cosine Transform (DCT). It is a stout process for image compression. It offers tremendous compaction for highly associated data. It gives good concession between material packing ability and computational complication. JPEG 2000 image compression procedure makes use of Discrete Wavelet Transform (DWT). DWT reduces the image size without degradation in the quality.
Face recognition finds a lot of claims in information security, human identification, security authentication, law implementation, smart cards, access mechanism and etc. Hence, it has drawn the attention of pattern recognition researchers.

3.1 IMAGE COMPRESSION

Image compression is classified into 2 types namely, Lossy and Lossless Image compressions.

- **Lossless Image Compression** - It is suitable for archiving and used in imaging, drawings and clip art.

- **Lossy** - It is specifically used at short bit rates to familiarize compression. They are best suited for natural images such as photographs, where unnoticeable loss of reliability achieves a extensive reduction in bit rate. Such compressions that produce indiscernible variances are called visually lossless compressions.

**Lossless Image Compressions**

- Run-length encoding - Evasion method in PiCture eXchange (PCX) and as one of possible ways in Bitmap (BMP), Truevision Graphics Adapter (TGA), Thousands of Incompatible File Formats (TIFF).

- Area image compression.
• Differential Pulse-Code Modulation (DPCM) and Predictive Coding.

• Entropy encrypting.

• Adaptive glossary procedures such as Lempel - Ziv - Welch (LZW) - used in Graphics Interchange Format (GIF) and TIFF.

• Deflation - used in PNG, Multiple-image Network Graphics (MNG) and TIFF.

• Series codes.

Lossy Compressions

• **Reduction of pigment space** - The pigment space is reduced to hold the most mutual flags in the image. The nominated colors are mentioned in the color palette of the compacted image’s header. Each pixel refers to the index of a color in the color palette.

• **Chroma subsampling** - The fact about the sharp perception of spatial changes of brightness of the human eye when compared to color is considered

• **Transform coding** - In this, a Fourier-related transform such as the Discrete Cosine Transform (DCT) (Ahmed et al 1974) is widely used.
• Fractal compression.

3.1.1 Properties of Image compression

There are diverse properties of Image compression as discussed below. They include:

• Scalability
• Region of interest coding
• Meta information
• Processing power

Scalability

Scalability, otherwise called progressive coding or embedded bit streams, deals with the reduction in image quality by manipulating the bit streams. It does not involve decompression and re-compression. Though its nature is contradictory, it is found in lossless codes, usually in form of pixels. It aids in promoting images while transferring them and provides variable quality access. There are numerous types of scalability as listed below.

• Quality progressive or layer progressive: The reconstructed image is refined by the bitstream.
- **Resolution progressive**: Initially, a lower image resolution is encoded.

- **Component progressive**: Initially grey is encoded, followed by color.

**Coding Regions of Importance**

Encoding the certain parts of the image with higher eminence than others is done. This can be combined with scalability.

**Meta information**

The compacted data contains evidence about images which can be used to group, search or browse images. Such information includes color, texture of data, small preview images and author or copyright information.

**Dispensation power**

The compression algorithms involves different processing power to encode and decode. Algorithms with High compression requires high processing power. Peak Signal-to-Noise Ratio (PSNR) determines the eminence of a compression is the extent of noise introduced by a lossless compression of the image.
**Merits of Image Compression**

Image compression aids users by loading pictures and webpages faster, using less space on a Web host. The physical size of an image is not reduced, but the data that makes up the image is compressed into a smaller size.

- **Size Reduction** - The main benefit of image compression is the fall of file size. The image can be compressed to whatever size needed. The image consumes less space on the hard drive but retains the same physical size, unless edited using an image editor. This helps the web developers to create their own image filled sites consuming less bandwidth and storage space.

- **Slow Devices** - Image compression helps easy loading of uncompressed pictures from devices like computers or cameras. For a website to be fully functional, compressed images help in faster loading of images. Compressed images can be loaded faster from a CD drive or a hard drive.

**Demerits of Image Compression**

There are some demerits in image compression.

- **Degradation** - Sometimes image degradation happens, meaning the quality of the image declines. The data is not lost but the quality of the image goes down.
• **Data Loss** - With some common file types, such as JPEG, the compression program discards some of the image's data permanently, when it shrinks in size. Before commencing compression, an uncompressed backup should be stored for restoration.

3.2 **PARTICLE SWARM OPTIMIZATION (PSO)**

Dr. Eberhart and Dr. Kennedy developed Particle Swarm Optimization (PSO), a population based stochastic optimization technique in 1995. It resembles some evolutionary computation practices like Genetic Algorithms (GA).

PSO is magnificently applied in many research application areas as it offers better, faster and cheaper results compared to other methods.

The system is initialized with a population of random solutions and optima is searched by apprising generations. In PSO, every resolution is likened unto a ‘bird’ in the search space called as ‘particle’. The particles are the possible solutions that fly through the delinquent space by following the current optimal particles.

• Each particle keeps record of the coordinates in the problem space that are associated with the best solution (pBest) accomplished until now along with the fitness values.
• When a particle takes all the population as its topological neighbors, the best value is represented as the gBest.

The particle swarm optimizer tracks another best value obtained so far by any particle in the neighbours of the particle called lBest. At each time step, the rate of hastening each particle is changed. A arbitrary term is used for assessing acceleration, with separate random numbers being generated for acceleration toward pBest and lBest locations.

PSO necessitates adjusting only few parameters. Hence, it facilitates simple modification for applications with specific requirement.

The Algorithm

The evolutionary techniques have the following procedure:

1. Random generation of the population.
2. Reckoning of a fitness value.
3. Reproduction of the population.
4. If requests are met, then stop. Otherwise go back to 2.

As already mentioned, PSO is based on the behaviours of bird flocking. Suppose a group of birds ponder for food in an area. Assume that there is some food in a place in the
area being searched. All the birds are not aware of the location of the food but they have an idea of how far the food is in each iteration. The best strategy for the bird to find the food is to follow the bird which is nearest to the food. Such policies are useful in solving the optimization problems.

The fitness of the elements has fitness values which are assessed by the ability function that is to be optimized. The algorithm is modified with a group of random particles (solutions) and then pursuits for optima by updating generations. In every iteration, each particle is updated by following two ‘best’ values - pBest and gBest. After finding the two best values, the particle updates its velocity and positions using the following Equations (3.1) and (3.2).

\[
\begin{align*}
\text{v[]} &= \text{v[]} + c1 \times \text{rand()} \times (\text{pBest}[] - \text{present}[]) + c2 \times \text{rand()} \times (\text{gBest}[] - \text{present}[]) \\
\text{present}[] &= \text{present}[] + \text{v}[]
\end{align*}
\]  

(3.1)

(3.2)

where,

- \(v[]\) - The particle velocity
- \(\text{present}[]\) - The current particle (solution)
- \(\text{pBest}[]\) and \(\text{gBest}[]\) - Best values
- `rand()` - Random number between (0,1).

- `c1, c2` - learning factors (`c1 = c2 = 2`)

---

**Figure 3.1 PSO Algorithm**

- Initialise population of particles with random positions and velocities
- Evaluate the fitness of each particle
- Compare each particle’s fitness with the current particle to obtain pBest
- Compare fitness evaluation with the population overall previous best to obtain GBest

\[
V_i = v_i + 2 \cdot \text{rand()} \cdot (p\text{best}_i - x_i) + 2 \cdot \text{rand()} \cdot (G\text{best}_i - x_i)
\]

\[
x_i = x_i + V_i
\]

- Is the stopping criteria?

  - NO
  - YES

- Stop
The algorithm keeps track of the following variables:

- Target value or condition
- Global best (gBest) value
- Stop value

```
For each particle
{
    Initialize particle
}

Do until maximum iterations or minimum error criteria
{
    For each particle
    {
        Calculate Data fitness value
        If the fitness value is better than pBest
        {
            Set pBest = current fitness value
        }
        If pBest is better than gBest
        {
            Set gBest = pBest
        }
    }

    For each particle
    {
        Calculate particle Velocity
        Use gBest and Velocity to update particle Data
    }
}
```

Figure 3.2 Pseudocode for PSO algorithm
Each particle consists of:

- Data representing a possible solution.
- Value of velocity indicating to what extent the data can be changed.
- A personal best (pBest) value indicating how closer the particle’s data has ever come to the Target.

In contrast to GA, PSO does not involve evolution operators such as edge and alteration. Elements apprise themselves with the inner velocity. Similarly, the information distribution appliance in PSO is significantly different. In GAs, chromosomes share evidence among themselves. So the whole population moves as a group in the direction of an optimum area. In PSO, only gBest stretches out the information to others. Hence, it is a one-way information sharing tool. In other words, it only looks for the superlative resolution. Compared to GA, in most cases, all the elements tend to congregate to the best resolution rapidly even in the local description.

3.3 DISCRETE COSINE TRANSFORM (DCT)

A Discrete Cosine Transform (DCT) is a finite series of data facts. DCTs are applied to numerous claims in Science and Engineering including lossy compression of audio (e.g. MP3) and images (e.g. JPEG), where small, high-frequency constituents can be discarded.
DCT, similar to the Discrete Fourier Transform (DFT) is a Fourier-related transform that uses actual numbers. It renovates a signal or image from the 3-D domain to the occurrence domain. DCTs are roughly twice the length of DFTs, operating on real data. DCT helps in separating the image into parts based on the image's visual quality.

To approximate a typical signal, cosine functions are used rather than sine functions as they are critical for compression. For differential equations, the cosines express a particular choice of frontier conditions.

There are eight typical DCT variants, out of which four are common. The most common irregular of DCT is the type-II DCT, which is often called simply ‘the DCT’ (Ahmed et al 1974, Rao & Yip 1990), its inverse, ‘the type-III DCT’ often called ‘the inverse DCT’ or ‘the IDCT’.

DCT is used in an image compression. In this, ‘N x N’ blocks are computed and the results are summarized. The formula is applied to each row and column of the block. The result is an 8 × 8 transform coefficient. The increasing vertical and horizontal index values represents higher vertical and horizontal spatial frequencies.

The input image is represented as N x M, ‘f(i,j)’ is the pixel intensity, I is the row and j is the column. ‘F(u,v)’ is the DCT coefficient in row ‘k₁’ and column ‘k₂’ of the DCT matrix.
The signal energy lies at low frequencies and appear in the upper left corner of the DCT. The lower right values represent higher frequencies, and are often small enough to be abandoned with little visible distortion. The DCT input is an ‘8 x 8’ array of integers. This array contains the gray scale level of each pixel. These pixels have levels from 0 to 255.

3.4 PROPOSED SYSTEM

A traditionally used approach to resolve discrete issues is to plot the discrete pursuit house to a constant area, apply a conventional PSO and then demap the result. This kind of plotting could be very simple or more refined. The equations of action contain the operators to perform the following actions.

- Calculating the change between two positions.
- Multiplying a velocity by using a geometric coefficient.
- Including two paces.
- Relating a rate to a position.

Face accessories include eyes, nose, mouth or facial templates reminiscent of nose size and width, mouth function and chin sort. These aspects are used to appreciate an unknown face by matching it to the closest neighbor in the stored database. Statistical
aspects extraction is typically pushed by means of algebraic ways equivalent to primary component analysis (PCA) and unbiased component evaluation (ICA) (Ramadan & Kader 2009). These methods find a mapping between the long-established characteristic areas to a slash dimensional feature space. The shortcoming of PCA is that it treats each the inner-class and out-courses equally (Zhao et al 2003, Shakhnarovich & Moghaddam 2004, Tolba et al 2006) and for that reason it's touchy to light and changes of expressions.

There are other replacement algebraic methods that are headquartered on transforms like down sampling, Fourier Transforms (feet), Discrete Cosine Transforms (DCT) and the Discrete Wavelet Transforms (DWT). Transformation centered feature extraction ways such because the DCT and DWT were determined to generate good FR accuracies involving very low computational fee (Hafed & Levine 2001). DCT is among the systems utilized in picture compressing which can be used to extract features (Matos 2008, Kekre 2010)

Wavelet analysis, a superb device in analysing unsteady alerts performs good in time area and frequency area.

The following steps are adopted.

- read the snap shots from database.
- Portraits are pre-processed.
• Compute 2d DCT of each photo.

• In finding the top of the line number of coefficients required utilizing Binary PSO.

• Retailer the optimized characteristic vector.

• Compute the Euclidean distance.

As a further section,

• Portraits are pre-processed.

• Read the test snapshot.

• Compute 2d DCT of test image.

The outcome from the above set of steps are processed.

• Declare the image comparable to minimal Euclidean distance as famous photo.
3.5 RESULTS AND DISCUSSION

There are 10 different images of 40 distinct subjects. For some of the subjects, the images are taken at different times, slightly varying lighting, facial expressions (open/closed eyes, smiling/non-smiling) and facial details (glasses/no-glasses). All the images are taken with a dark homogeneous background and the subjects are in upright and frontal position (with tolerance for some side movement).

The files are in PGM format and can be conveniently viewed using the 'xv' program. The size of each image is 92 x 112, 8-bit grey levels. The images are organised in
40 directories (one for each subject) named as ‘sX’, where ‘X’ indicates the subject number (between 1 and 40).

Table 3.1 ORL Face Database

<table>
<thead>
<tr>
<th>Database</th>
<th>PPR1</th>
<th>PPR2</th>
<th>PPR3</th>
<th>PPR4</th>
<th>No. of Trainings</th>
<th>No. of Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORL</td>
<td>impyramid</td>
<td>imadjust</td>
<td>Edge</td>
<td>imadd</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Referring to above table, we define the terms as follows:

- **Impyramid (I, direction):** Impyramid \((I, \text{ direction})\) computes a Gaussian pyramid reduction or expansion of ‘I’ by one level. The direction can be ‘reduced’ or ‘expanded’.

- **Imadjust\((\text{I}, [\text{lowin}; \text{highin}], [\text{lowout}; \text{highout}], \text{gamma})\):** \(J = \text{imadjust}(\text{I}, [\text{lowin}; \text{highin}], [\text{lowout}; \text{highout}], \text{gamma})\) maps the values in ‘I’ to new values in ‘J’, where gamma specifies the shape of the curve describing the relationship between the values ‘I’ and ‘J’. If gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values. If the argument is omitted, gamma defaults to 1 (linear mapping).
• **Edge(I,canny):** \( J = \text{Edge}(I) \) takes a gray scale or a binary image ‘I’ as its input and returns binary image ‘J’ of the same size as ‘I’, with 1's where the function finds edges in ‘I’ and 0's elsewhere. **Edge (I,'canny')** specifies the Canny method.

• **Uint8 (I):** \( \text{Uint8}(I) \) returns the stored integer value of an object as a built-in uint8. If the stored integer word length is too big for a uint8, or if the stored integer is signed, the returned value saturates to uint8.

• **Imadd(I1,I2):** \( Z = \text{Imadd}(X,Y) \) adds each element in array ‘X’ with the corresponding element in array ‘Y’ and returns the sum in the corresponding element of the output array ‘Z’. ‘X’ and ‘Y’ are real, non-sparse numeric arrays with the same size and class, or ‘Y’ is a scalar double. ‘Z’ has the same size and class ‘X’, unless ‘X’ is logical, in which case ‘Z’ is double.

• **Fspecial (‘sobel’):** \( h = \text{Fspecial('sobel')} \) returns a 3-by-3 filter ‘h’ that emphasizes horizontal edges using the smoothing effect by approximating a vertical gradient. If it is needed to emphasize vertical edges, transpose the filter ‘h’.

• **Imfilter (I, H, ‘same’):** \( \text{Imfilter} (I, H, ‘same’) \) filters the multidimensional array ‘I’ with the multidimensional filter ‘H’. The array ‘I’ can be logical or a non-sparse numeric array of any class and dimension. The result has the
same size and class as ‘I’. ‘Same’ means the output array is of the same size as the input array. This is the default behavior when no output size options are specified.

In each directory there are 10 different images of the selected subject named as ‘Y.pgm’, where ‘Y’ indicates the image for the specific subject (between 1 and 10).

Table 3.2 Experiments and Results

<table>
<thead>
<tr>
<th>DCT</th>
<th>20 * 20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unmodified</td>
</tr>
<tr>
<td>Recognition</td>
<td>92.5</td>
</tr>
<tr>
<td>Non-Zero Coefficient</td>
<td>218</td>
</tr>
</tbody>
</table>