CHAPTER 2

Gaussian Pyramid Compression

An image pyramid is a hierarchical representation of an image and is a data structure designed to support efficient scaled convolution through reduced image representation. It consists of a sequence of copies of an original image in which both sample density and resolution are decreased in regular steps. The original image can be exactly reconstructed from its pyramidal representation; hence the pyramid code is complete.

The pyramid representation can be used in data compression. Although it has one-third more sample elements than the original image, the values of these samples are close to zero, so that they can be represented with minimum number of bits. The pyramid offers a useful image representation for a number of tasks like image compression, image enhancement, texture analysis, image mosaicing, etc.

The Gaussian Pyramid, one of the pyramid varieties is a sequence of low-pass, down sample images; Gaussian Pyramids are efficient to compute coarse-scale images. Gaussian Pyramid construction is equivalent to convolving the original image with a set of Gaussian-like weighting functions.

The Gaussian Pyramid can be used for multi-scale edge estimation. The Gaussian Pyramid decomposes the image into a set of low pass filtered images. The generation of a three level Gaussian pyramid is illustrated in the fig. 2.1.
Figure 2.1: Generation of Gaussian Pyramid

The first step in pyramid coding is to utilize the low-pass filter on the original image $I_0$ to obtain image $I_1$. The $I_1$ is a reduced version of $I_0$, in that both resolution and sample density are decreased. In a similar way $I_2$ is a reduced version of $I_1$ and so on.

Filtering is performed by a procedure equivalent to convolution with one of a family of local, symmetric weighting functions. An important member of this family resembles the Gaussian probability distribution, so the sequence of images $I_0, I_1, I_2, \ldots I_n$ is called the Gaussian Pyramid.

2.1 Introduction

In this chapter, we investigate that the effect of Gaussian Pyramid Compression technique on the ability of an iris recognition system is accurately used to identify individuals. This compression technique is used to compress the eye image and this compressed eye is used for the localization of the inner and outer boundaries of the iris region. Located iris is extracted from the compressed eye image and after normalization and enhancement it is represented by a data set.

With Gaussian Pyramid Compression, improved matching performance is observed down to 0.25 bits/pixel (bpp), attributed to noise reduction without a significant loss of texture in order to conform, that, the iris matching algorithms are not degraded by image compression.
The performance is evaluated by means of the change in the Hamming Distances between iris codes using an iris recognition implementation based on Libor Masek’s algorithm [Lib05] and achieved high accuracy of 96%.

We seek to compress the original images because it is the data that is valuable and serves as training and testing images for the development of new algorithms. Typically, a database for an iris recognition system does not contain actual iris images, but rather it stores the compressed images stored as 512 bytes per eye.

Compression has been investigated and used in some biometric applications such as the FBI standard for fingerprint compression [Bur81; CAS04] using MPEG compression [Dau97; WC97] for video that may be used in facial recognition applications. There has been some limited research in the area of iris image compression [Dau97] but this compression applied to iris codes, not iris images. Here, Gaussian pyramid compression is applied to the iris imagery itself.

2.2 I R System

The entire Iris Recognition system flow is briefly described as follows. First, the eye image is captured using a standard camera. Image pre-processing module employs some image processing algorithms to demarcate the region of interest (i.e., iris zone) from the input image containing an eye. It performs four major tasks including iris compression, localization, normalization and enhancement. The compression technique is used to compress the eye image and this compressed eye is used for the localization of the inner and outer boundaries of the iris region.

The iris localization procedure segments the annular iris region from the entire eye image. First, it finds the circular inner (iris-pupil boundary) and outer circle (iris-sclera boundary) of the iris using a circular Hough transform. Then it marks the region of the annular iris ring that is not visible due to eyelids and eyelashes. The method uses a linear Hough transform to find the eyelids and a simple thresholding technique to find eyelashes covering the iris image.
The iris normalization procedure transforms the segmented iris region into a rectangular grid of 240 x 20 pixels using a polar-to-Cartesian transformation and bilinear interpolation. To minimize the effect on encoding of corrupted regions detected during segmentation, the method sets the intensity of corrupted regions to the neighborhood’s mean intensity.

Next, the iris feature extraction module encodes the normalized iris image as a binary string. At each pixel in the 240 x 20 image, it extracts information about local texture characteristics as described by convolution with two dimensional Gabor-wavelet filtering technique, whose real and imaginary components form a quadrature pair to generate the iris feature code.

The following Daugman’s method discards the amplitude of the complex-valued response [Dau01] and quantizes the phase so that only its quadrant in the complex plane is retained using two bit gray code. The feature vector is a 9600 bit template (240 x 20 x 2), which is combined with the corrupted bit mask computed in the segmentation stage.

Finally, the iris pattern recognition module employs a minimum distance classifier according to Hamming Distance to recognize the iris pattern by comparing the iris code with the enrolled iris codes in the iris code database.

2.2.1 Image Preprocessing

A good and clear image eliminates the process of noise removal and also helps in avoiding errors in calculation. In this stage, we transformed the images from RGB to gray level for further processing. Before extracting features from the original image, the image needs to be preprocessed to compress the image; localize and normalize iris and reduce the influence of the factors such as brightness, non-uniform illumination, etc., such preprocessing is described in the following subsections.
**Gaussian Pyramid Compression**

The original image is convolved with a Gaussian kernel. The resulting image is a low-pass filtered version of the original image. Suppose the image is represented initially by the array $g_0$, which contains $C$ columns and $R$ rows of pixels. Each pixel represents the light intensity at the corresponding image point by an integer $I$ between $0$ and $k-1$ where $k$ is the maximum gray level value. This image becomes the bottom or zero level of the Gaussian pyramid.

Pyramid level 1 contains image $g$, which is reduced or low-pass filtered version of $g_0$. Each value within level 1 is computed as a weighted average of values in level 0 within a 2 by 2 window. Each value within level 2 representing $g_2$ is obtained from values within level 1 by applying the same pattern of weights.

![Gaussian Pyramid Diagram](image)

**Figure 2.2: Gaussian Pyramid**

A graphical representation of this process in one dimension, which generates a Gaussian pyramid, is shown in the fig. 2.2.
Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern of “generating kernel” is used to generate all levels. The size of the weighting function is not critical [Bur81].

Gaussian Pyramid construction is equivalent to convolving the original image with a set of Gaussian-like weighting functions and decomposes the image into a set of low pass filtered images. Fig. 2.3 shows first two levels of the Gaussian pyramid for the iris image. The original image level 0 measures 255 X 255 pixels and each higher-level array is roughly half the dimension of its predecessor. Thus, level 2 measures just 64 X 64 pixels.

![Figure 2.3: Low pass filter effect of the Gaussian pyramid, which is of the order of 1.7:1 for RGB image and 1.9:1 for gray image](image)

(a) 255 X 255 pixels (11.582 KB)  
(b) 128 X 128 pixels  
(c) 64 X 64 pixels (5.6572 KB)
We have selected the 2-by-2 patterns because it provides adequate filtering at low computational cost. The low-pass filter effect of the Gaussian pyramid, which is of the order of 1.7:1 for RGB image and 1.9:1 for gray image, is clearly shown in the Fig. 2.3.

**Iris Localization**

Before performing iris pattern matching, the image is compressed by Gaussian pyramid, and then the boundaries of the iris are to be located. The detailed step by step iris localization process is as follows.

**STEP 1:** First the captured image must be converted to grayscale format if it is not already in an intensity image. This is done by subtracting each pixel’s intensity of image from 255 and gives the complement of image.

**STEP 2:** Apply Gaussian Pyramid technique to compress the eye image so as to reduce the size from 255 x 255 pixels to 64 x 64 pixels.

**STEP 3:** Apply scaling function to the complemented compressed eye image which reduces the complexity significantly and scaling down the images to the constant size.

**STEP 4:** Filling holes in the intensity image. A hole is an area of dark pixels surrounded by lighter pixels. Flood fills the foreground region of the gray image to eliminate any small holes caused by specular and corneas reflection by using morphological operation.

**STEP 5:** Project the complemented image in the vertical and horizontal direction to approximately estimate the centre co-ordinates \((x_p, y_p)\) of the pupil. Since the pupil is generally darker than its surroundings, the co-ordinates corresponding to the minima of the two projection profiles are considered as the centre co-ordinates of the pupil.

**STEP 6:** By adaptively selecting a reasonable threshold using gray-level histogram of this region, binarized region is centered at the point \((x_p, y_p)\). The centroid of the resulting binary region is considered as a more accurate estimate of the pupil co-ordinates and roughly computes the radius of the pupil.

**STEP 7:** Apply canny edge detection and circular Hough transform to certain region of the edge image to estimate the exact parameters of inner and outer circles of iris.
These parameters are for pupil \((x_{p0}, y_{p0})\) is the center of the pupil, \(r_p\) is the radius of the pupil circle, for outer circle \((x_{i0}, y_{i0})\) is the center and \(r_i\) is the radius.

Integro-differential operator is used for locating the inner and outer boundaries of iris as shown in the fig. 2.4. The operator computes the partial derivative of the average intensity of circle points with respect to increasing radius \(r\).

![Figure 2.4 Compressed Image with Inner and Outer boundary circles](http://www.novapdf.com/)  

After convolving the operator with Gaussian kernel, the maximum difference between inner and outer circle will define the center and radius of the iris boundary. For upper and lower eyelids detection, the path of contour integration is modified from circular to parabolic curve.
The operator is accurate because it searches over the image domain for the global maximum. It can compute at faster rate because it uses the first derivative information.

**Iris Normalization and Enhancement**

Irises from different people may be captured in different sizes and even for the iris from the same person; the size may change because of the variation of the illumination and other factors. Such elastic deformations in iris texture affect the results of iris matching.

![Figure 2.5: Normalization Process](image)

Figure 2.5: Normalization Process

For the purpose of achieving more accurate recognition results, the homogenous rubber sheet model devised by Daugman [Dau04] used to remap each point within the iris region to a pair of polar co-ordinates \((r, \theta)\) where \(r\) is the interval \([0, 1]\) and \(\theta\) is the angle \([0, 2\pi]\).

The rubber sheet model is shown in fig 2.5, which takes into account pupil dilation and size inconsistencies in order to produce a normalized representation with constant dimensions.
The normalized iris image has low contrast and non-uniform illumination caused by the light source position. The image needs to be enhanced to compensate for these factors. The local histogram analysis is applied to the normalized iris image to reduce the effect of non-uniform illumination and obtain well-distributed texture image. Reflection regions are characterized by the high intensity values close to 255. A simple thresholding operation is used to remove the reflection noise.

2.2.2 Feature Extraction

Gabor transform is used for analyzing the human iris patterns and extracting feature points from them. In order to extract the discriminating features from the normalized collarette region, the normalized pattern is convolved with 1D Gabor wavelet [Lib05]. Thus, feature encoding is implemented by first breaking the two-dimensional normalized iris pattern into one-dimensional wavelets and then these signals are convolved with 1D Gabor wavelet.

The resulting phase information for both the real and imaginary response is quantized, generating a bit-wise template. In this work, the angular and radial resolutions are set as 240 and 20 pixels, respectively. The two bits are used to represent the quantized phase information for each pixel. Therefore, the total size of the iris template is 9600 bits.

2.2.3 Template Matching

Template matching is performed using the Hamming Distance measure, taking into account the occluded regions defined as masks, by not comparing the feature points extracted from these regions. The result is the number of bits that are different between the binary codes in the non-occluded regions, divided by the number of bits compared.
The Hamming Distance is the matching metric employed by Daugman and calculation of the Hamming Distance is taken only with bits that are generated from the actual iris region. The decision of whether these two images belong to the same person depends upon the following result [Dau04].

- If $HD = 0$, infer that it is a perfect match between two iris codes
- If $HD \leq 0.32$, infer that it is same iris
- If $HD > 0.32$, infer that it is different iris

Hamming distance of $\leq 0.32$ allows identification with high confidence and used as a threshold for recognition. To derive the performance results, each original iris image was compared against every other image in the database.

2.3 Experimental Results

Experiments are conducted on CASIA iris database provided by the Chinese Academy of Sciences Institute of Automation [CAS04] and 135 images are chosen for the experiments. Those 135 images are divided into 27 classes and each of them had 5 images. The first image from each class is selected as the template. The remaining 4 images of each class are adopted as the test set.

2.3.1 Recognition Analysis

We reached the correct segmentation for the pupil center and outer boundary radius and also the average execution time for inner boundary detection was 0.016s, 0.031s for outer boundary detection and for matching was 0.016s using Matlab 7. The statistical distribution of Hamming distance between different and same iris feature vectors for both compressed and uncompressed iris is shown in fig. 2.6 and fig.2.7.

The y-axis and x-axis indicate the number of data samples and the Hamming distance respectively. It indicates that the Hamming distance is between 0.16 and 0.32 for the iris images from the same eye and between 0.33 and 0.55 for the iris images.
from different eyes. The Hamming distance of sample data for both compressed and uncompressed irises are shown in table 2.1.

**Table 2.1: Iris sample data for same/different person**

<table>
<thead>
<tr>
<th>Same / Different Iris</th>
<th>Hamming Distance (HD)</th>
<th>Uncompressed Iris</th>
<th>Compressed Iris</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris1L1 Iris1L2</td>
<td>0.2142</td>
<td>0.1945</td>
<td></td>
</tr>
<tr>
<td>Iris1L1 Iris1L3</td>
<td>0.1536</td>
<td>0.1893</td>
<td></td>
</tr>
<tr>
<td>Iris3L1 Iris3L2</td>
<td>0.2069</td>
<td>0.1796</td>
<td></td>
</tr>
<tr>
<td>Iris3L1 Iris3L5</td>
<td>0.2444</td>
<td>0.2558</td>
<td></td>
</tr>
<tr>
<td>Iris2L2 Iris2L4</td>
<td>0.2027</td>
<td>0.2447</td>
<td></td>
</tr>
<tr>
<td>Iris2L3 Iris2L5</td>
<td>0.2302</td>
<td>0.2164</td>
<td></td>
</tr>
<tr>
<td>Iris1L1 Iris1R1</td>
<td>0.4275</td>
<td>0.4052</td>
<td></td>
</tr>
<tr>
<td>Iris1L1 Iris1R5</td>
<td>0.4621</td>
<td>0.4435</td>
<td></td>
</tr>
<tr>
<td>Iris2L1 Iris2R3</td>
<td>0.4878</td>
<td>0.4859</td>
<td></td>
</tr>
<tr>
<td>Iris2L1 Iris2R4</td>
<td>0.4860</td>
<td>0.4796</td>
<td></td>
</tr>
<tr>
<td>Iris3L1 Iris3R5</td>
<td>0.4950</td>
<td>0.4534</td>
<td></td>
</tr>
</tbody>
</table>

L1 denotes left eye, R denotes right eye, number denotes before L or R are different eyes, number denotes after L or R are same eye with different illumination. For example Iris1L1 and Iris1L3 belongs to same person 1, but with different light-effect, it considered to be different images and matching process proceeds.

The graphical representation in fig 2.8 shows the recognition rate according to the Hamming distance. Table 2.2 shows the different Hamming distance to find the recognition rate of an iris. The correct recognition rate of this system is 96% when we use 27 classes (135 images). The processing time of iris recognition is relatively lesser when compared to the state of art methods are shown in fig. 2.9 and fig. 2.10. From the experimental results, the proposed method shows an overall accuracy of 96%.
Figure 2.6: The distribution of Hamming Distance for Compressed Iris

Figure 2.7: The distribution of hamming distance for Uncompressed Iris
Iris Recognition Rate Using Hamming Distance

Figure 2.8: Iris Recognition Rate using Hamming Distance
Figure 2.9: Processing Time of Compressed Vs Uncompressed for same Iris
Figure 2.10: Processing Time of Compressed Vs Uncompressed for different Iris
Table 2.2: Recognition Rate According to the Hamming Distance

<table>
<thead>
<tr>
<th>HD</th>
<th>Compressed</th>
<th>Compressed (%)</th>
<th>Uncompressed (135 Images)</th>
<th>Uncompressed (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.28</td>
<td>128</td>
<td>95</td>
<td>125</td>
<td>92</td>
</tr>
<tr>
<td>0.29</td>
<td>130</td>
<td>96</td>
<td>126</td>
<td>93</td>
</tr>
<tr>
<td>0.30</td>
<td>130</td>
<td>96</td>
<td>126</td>
<td>93</td>
</tr>
<tr>
<td>0.31</td>
<td>130</td>
<td>96</td>
<td>127</td>
<td>94</td>
</tr>
<tr>
<td>0.32</td>
<td>130</td>
<td>96</td>
<td>127</td>
<td>94</td>
</tr>
<tr>
<td>0.33</td>
<td>130</td>
<td>96</td>
<td>129</td>
<td>96</td>
</tr>
<tr>
<td>0.34</td>
<td>130</td>
<td>96</td>
<td>131</td>
<td>97</td>
</tr>
<tr>
<td>0.35</td>
<td>131</td>
<td>97</td>
<td>131</td>
<td>97</td>
</tr>
<tr>
<td>0.36</td>
<td>132</td>
<td>98</td>
<td>131</td>
<td>97</td>
</tr>
</tbody>
</table>

2.4 Summary

The proposed iris recognition method is relatively simple and efficient against the existing methods. We reached the correct segmentation for the pupil center and outer boundary radius and also the average execution time for inner boundary detection was 0.016s, 0.031s for outer boundary detection and for matching it was 0.016s. From the experimental results, the proposed method shows an overall accuracy of 96%. The processing time of iris recognition is relatively lesser when compared to the existing methods. It is observed that the result is not perfect due to low quality of the iris images. The iris region is occluded by eyelids and eyelashes. The experimental results show that the proposed approach has a good recognition performance and speed. This experiment is applicable to do experiments on a larger iris database in various environments for iris recognition system.