Iris Recognition: Gabor Filtering[*]

David Carr

Abstract

Gabor wavelets, filtering and Hamming distance

1 Gabor Wavelets

To understand the concept of Gabor filtering, we must first start with Gabor wavelets. Gabor wavelets are formed from two components, a complex sinusoidal carrier and a Gaussian envelope.

$$g(x,y) = s(x,y)\, w_r(x,y)$$

The complex carrier takes the form:

$$s(x,y) = e^{j(2\pi(u_0x+v_0y)+P)}$$

We can visualize the real and imaginary parts of this function seperately as shown in this figure.

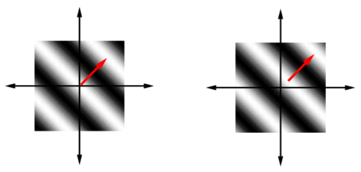

Figure 1: Image Source[1] (a) Real Part (b) Imaginary Part

The real part of the function is given by:

$$Re(s(x,y)) = cos(2\pi(u_0x+v_0y)+P)$$

and the imaginary:

$$Im(s(x,y)) = sin(2\pi(u_0x+v_0y)+P)$$

*Version 1.4: Dec 18, 2004 6:00 pm US/Central
†http://creativecommons.org/licenses/by/1.0
[1]http://mplab.ucsd.edu/tutorials/pdfs/Gabor.pdf

The parameters u_0 and v_0 represent the frequency of the horizontal and vertical sinusoids respectively. P respresents an arbitrary phase shift.

The second component of a gabor wavelet is its envelope. The resulting wavelet is the product of the sinusoidal carrier and this envelope. The envelope has a gaussian profile and is described by the following equation:

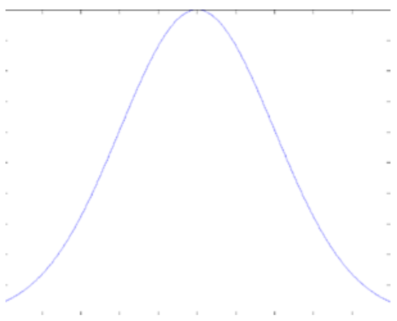

Figure 2: Gaussian Envelope

$$g(x,y) = Ke^{-\pi\left(a^2(x-x_0)_r^2+b^2(y-y_0)_r^2\right)}$$

where:

$$(x-x_0)_r = (x-x_0)\,cos(\theta) + (y-y_0)\,sin(\theta)$$

$$(y-y_0)_r = -(x-x_0)\,sin(\theta) + (y-y_0)\,cos(\theta)$$

The parameters used above are K - a scaling constant (a,b) - envelope axis scaling constants, θ - envelope rotation constant, (x_0,y_0) - Gausian envelope peak.

To put it all together, we multiply $s(x,y)$ by $w_r(x,y)$. This produces a wavelet like this one:

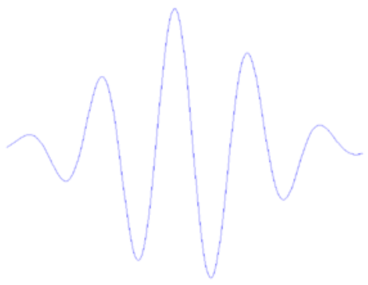

Figure 3: 1D Gabor Wavelet

2 Generating an Iris Code

Now that we have Gabor wavelets, lets do something interesting with them. Lets start with an image of an eye and then unroll it (map it to cartesian coordinates) so we have something like the following:

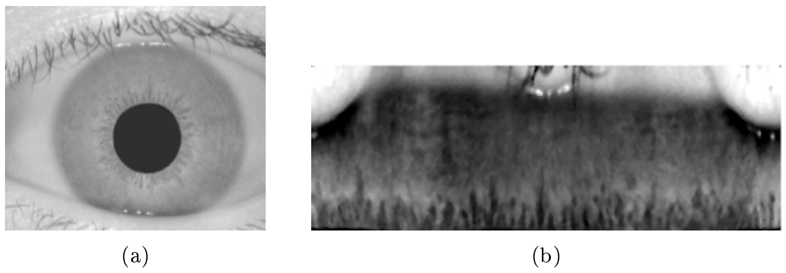

Figure 4: (a) Image of Eye (b) "Unrolled" Iris

What we want to do is somehow extract a set of unique features from this iris and then store them. That way if we are presented with an unknown iris, we can compare the stored features to the features in the unknown iris to see if they are the same. We'll call this set of features an "Iris Code."

Any given iris has a unique texture that is generated through a random process before birth. Filters based on Gabor wavelets turn out to be very good at detecting patterns in images. We'll use a fixed frequency 1D Gabor filter to look for patterns in our unrolled image. First, we'll take a one pixel wide column from our unrolled image and convolve it with a 1D Gabor wavelet. Because the Gabor filter is complex, the result will have a real and imaginary part which are treated seperately. We only want to store a small number of bits for each iris code, so the real and imaginary parts are each quantized. If a given value in the result vector is greater than zero, a one is stored; otherwise zero is stored. Once all the columns of the image have been filtered and quantized, we can form a new black and white image by putting all of the columns side by side. The real and imaginary parts of this image (a matrix), the iris code, are shown here:

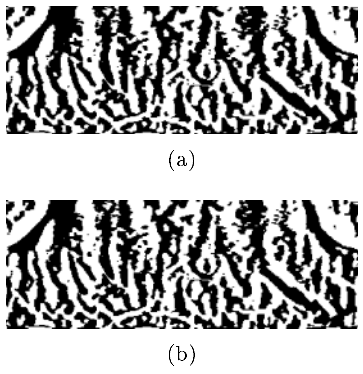

Figure 5: (a) Real Part of Iris Code (b) Imaginary Part of Iris Code

Now that we have an iris code, we can store it in a database, file or even on a card. What happens though if we want to compare two iris codes and decide how similar they are?

3 Comparing Iris Codes

The problem of comparing iris codes arises when we want to authenticate a new user. The user's eye is photographed and the iris code produced from the image. It would be nice to be able to compare the new code to a database stored codes to see if this user is allowed or to see who they are. To perform this task, we'll attempt to measure the Hamming distance between two iris codes. The Hamming distance between any two equal length binary vectors is simply the number of bit positions in which they differ divided by the length of the vectors. This way, two identical vectors have distance 0 while two completely different vectors have distance 1. Its worth noting that on average two random vectors will differ in half their bits giving a Hamming distance of 0.5. The Hamming distance is mathematically defined in this equation:

$$D = \frac{A \oplus B}{\text{length(A)}}$$

In theory, two iris codes independently generated from the same iris will be exactly the same. In reality though, this doesn't happen vary often for reasons such as imperfect cameras, lighting or small rotational errors. To account for these slight inconsistencies, two iris codes are compared and if the distance between them is below a certain threshold we'll call them a match. This is based on the idea of statistical independance. The iris is random enough such that iris codes from different eyes will be statistically independent (ie: have a distance larger than the threshold) and therefore only iris codes of the same eye will fail the test of statisical independance. Empirical studies with millions of images have supported this assertion. In fact, when these studies used the threshold used in our method (.3) false positive rates fell below 1 in 10 million.