
4.1. Overview of Phase-II
In the previous chapter it has been shared that initial phase of DMT scheme was focused upon the HVS modeling using DCT and Fuzzy Inference system. To integrate all the three sensitive areas identified and extracted from a gray scale image and its implementation using MATLAB coding. But, further the research work to achieve objective of this research work need to identify the hardware implementation of the DMT scheme.

4.2. Phase-II (Simulation of Image Watermarking Model)
Furthermore, the research work was redirected towards the simulation of image watermarking model. It required the steps for implementation which were already taken care from previous implementations. Now, the Block Transforming Techniques used in this research work used following block transformation techniques Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT), needed to be implemented during transformation of an image from its spatial domain to frequency domain so, that the image watermarking could be robust and can with stand the tampering with the original image, by attacks applied for its removal.

i) Discrete Cosine Transform (DCT): A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (fewer functions are needed to approximate
a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions.

**Equation 4-1: DCT for differential equations**

\[
y(k, l) = w(k) \sum_{m=1}^{M} u(m, l) \cos \left( \frac{\pi (2m - 1)(k - 1)}{2M} \right), k = 1, \ldots, M
\]

Where

\[
w(k) = \begin{cases} 
\frac{1}{\sqrt{M}}, & k = 1 \\ 
\frac{2}{\sqrt{M}}, & 2 \leq k \leq M 
\end{cases}
\]

DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

ii) Discrete Wavelets Transform (DWT): In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time).

The DWT of a signal \( x \) is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse response ‘\( g \)’ resulting in a convolution of the two:

**Equation 4-2: DWT Equation**

\[
y[n] = (x \ast g)[n] = \sum_{k=-\infty}^{\infty} x[k] g[n-k]
\]
The mentioned techniques are frequently used in the field of steganography, cryptography, watermarking etc; which are frequently used for data authentication. The proposed work relates to the field of image authentication through digital watermarking technique which is planned to be implemented in the real time domain. The data fuzzification and neural networks are the techniques which are readily used for such purpose. The research work has demonstrated the implementation of real-time robust digital watermarking, by using these techniques.

4.3. BlueThresh() Thresholding Method

This method in the research work propose a technique to embed imperceptible watermark in an image in real-time. The model constitutes of webcam needed to acquire an image in realtime, Matlab version 8.0 with Simulink running on a computer. The acquired image constitutes of RGB colour frames; the work propose to extract the Blue frame for embedding the watermark in it and merge it with other two Red and Green colour frames to reconstitute the image. This process makes the image watermarking robust and optimized.

Blue Channel selected from the RGB Image and applying the formulation to find the threshold that minimizes the weighted within-class variance.

This turns out to be the same as maximizing the between-class variance.

Operates directly on the gray level histogram \([e.g. 256 \text{ numbers}, P(i)]\), so it’s fast (once the histogram is computed).

Histogram (and the image) are bimodal.

No use of spatial coherence, nor any other notion of object structure.

Assumes stationary statistics, but can be modified to be locally adaptive. (exercises)
Assumes uniform illumination (implicitly), so the bimodal brightness behavior arises from object appearance differences only.

The weighted within class variance is:

\begin{equation}
\text{Equation 4-3: Class Variance}
\end{equation}

\[
\sigma^2_W(t) = q_1(t)\sigma^2_1(t) + q_2(t)\sigma^2_2(t)
\]

Where, the class probabilities are estimated as:

\begin{equation}
\text{Equation 4-4: Class Probability}
\end{equation}

\[
q_1(t) = \sum_{i=1}^{t} P(i)
\]

\[
q_2(t) = \sum_{i=t+1}^{T} P(i)
\]

And the class means are

\begin{equation}
\text{Equation 4-5: Class Means}
\end{equation}

\[
\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)}
\]

\[
\mu_2(t) = \sum_{i=t+1}^{T} \frac{iP(i)}{q_2(t)}
\]

Finally, the individual class variances are:

\begin{equation}
\text{Equation 4-6: Individual Class Variance}
\end{equation}

\[
\sigma^2_1(t) = \sum_{i=1}^{t} \left[ i \mu_1(t) \right]^2 \frac{P(i)}{q_1(t)}
\]

\[
\sigma^2_2(t) = \sum_{i=t+1}^{T} \left[ i \mu_2(t) \right]^2 \frac{P(i)}{q_2(t)}
\]

This process run through the full range of \( t \) values [1,256] and pick the value that minimizes the value of \( \sigma^2_W(t) \).
But the relationship between the within-class and between class variances can be exploited to generate a recursion relation that permits a much faster calculation.

For any given threshold, the total variance is the sum of the within-class variances (weighted) and the between class variance, which is the sum of weighted squared distances between the class means and the grand mean, the total variance can be expressed as:

\[ \sigma^2 = \sigma_w^2(t) + q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2 \]

With-in class from before \( \sigma_w^2(t) \)

Between class, \( \sigma_B^2(t) \) is \( q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2 \)

Since the total is constant and independent of \( t \), the effect of changing the threshold is merely to move the contributions of the two terms back and forth.

So, minimizing the within-class variance is the same as maximizing the between-class variance.

The nice thing about this is that it can compute the quantities in \( \sigma_B^2(t) \) recursively as it run through the range of \( t \) values.

Finally,

\[ q_1(1) = P(1) ; \mu_1(0) = 0 \]
\[ q_1(t + 1) = q_1(t) + P(t + 1) \]
\[ \mu_1(t + 1) = \frac{q_1(t)\mu_1(t) + (t + 1)P(t + 1)}{q_1(t + 1)} \]
\[ \mu_2(t + 1) = \frac{\mu q_1(t + 1)\mu_1(t + 1)}{1 - q_1(t + 1)} \]
The model depicted below is used in the research work for simulating the real-time watermark embedding in the blue channel of an image captured through webcam.

**Figure 4-1: SIMULINK MODEL FOR REAL-TIME WBC**

The profile summary shows the time consumed in the entire process of real-time image watermarking using the proposed method.

**Figure 4-2: PROFILE SUMMARY OF SIMULINK BASED REAL-TIME WBC**
Following are the result of watermark embedding process adopted in this paper using SIMULINK.

**Figure 4-3: Original Image Captured in Realtime**

**Figure 4-4: Watermarked Image in Blue Channel**

MSE: 4.2053dB; PSNR: 41.957dB

**4.4. Conclusion of Simulink based real-time watermark embedding**

Computed value of SIM(X, X*) parameter for the image depicted in Figure 4-4 (Singed Image) is 18.5987 which indicates a good watermark recovery process. The time consumed in image watermarking is computed as approx. 12 seconds, this model could be extended for the realtime digital image watermarking in camera enabled mobile devices for improving the authenticity of images captured and shared using Smartphone.