LINEAR ALGEBRA

IMAGE COMPRESSION AND WAVELET TRANSFORMATION

Raman Shukla
2020201098
Mtech CSE

Introduction

The aim of any image compression is to remove the redundancies for the better use of that image. There are mainly two types of compression, namely Lossless and Lossy. We are focusing here on the lossy image compression, as it manages to change the size of the image significantly with some loss of information.

The wavelet transform is the primary technique used for this type of image compression with little variations.

I have written my report on the subtopic **wavelet transformation.**

Image Compression In JPEG

The role of data compression is to convert strings of bits that represent data into shorter and more economical ways of storage and processing. In the early 1970s, most of the compression techniques were based on lossless compression. These techniques yield a low compression ratio for continuous tone images. This was primarily because the pixels tend to spread out over the entire range i.e. there were a few high frequency peaks in the plotted histogram.
Why compress images? Consider a colour picture with 400 dpi. If the image is 15inches2, we will have 2,400,000 pixels, since 3 bytes will be required to store color information, it will account for approximately 7MB of memory. These kinds of images will need more transfer time when sent electronically. In lossless compression like zip, tar files, the image quality is

not sacrificed. JPEG uses the baseline algorithm for lossy compression. JPEG is a format for image compression based on the discrete cosine transformation such that some of the data is lost during the compression process, but the loss is not easily perceived by the human eye. . The baseline algorithm is capable of compressing tone images to less than 10% of their original sizes. These algorithms take advantage of known limitations of the human eye.

JPEG (Joint Photographic Experts Group) is a format for image compression based on the discrete cosine transform. The JPEG lossy compression algorithm consists of three stages:

1. DCT Transformation
2. Coefficient Quantization
3. Lossless Compression

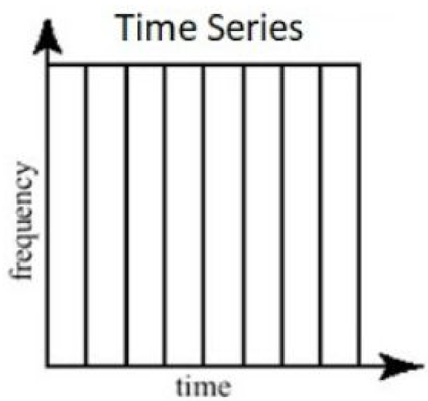
The first and last stages of compression are mainly lossless procedures, since negligible precision is lost during DCT transformation. The 2nd stage is the driving force behind JPEG's compression ratio. In this step most of the insignificant data is disregarded.

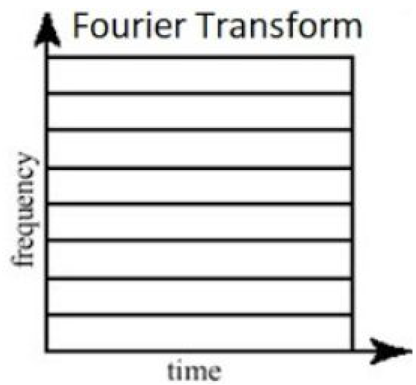
Wavelet Transformation

To go further first we need to understand Fourier Transform (FT). Any raw signal is generally a time domain signal. But we sometimes need other values hidden in the signal, i.e the frequency spectrum of the signal. FT (Fourier Transform) changes a time-domain signal to a frequency domain signal. Thus, we use transformation for changing the domain. FT changes a time-domain signal to a frequency domain signal.

FT just like Wavelet Transform (WT) is a reversible transform. We can transform the signal from raw to processed form back and forth. But we do not have frequency information in raw signal and time-domain information in the processed signal. WT on the other hand transforms the signal in both time-frequency domains. Although not as precise as the FT and raw signal are in their respective domains.

The other limitation of FT is with non stationary signals, where frequency component changes with time.





We can see above that how our function changes from good time resolution to good frequency resolution.

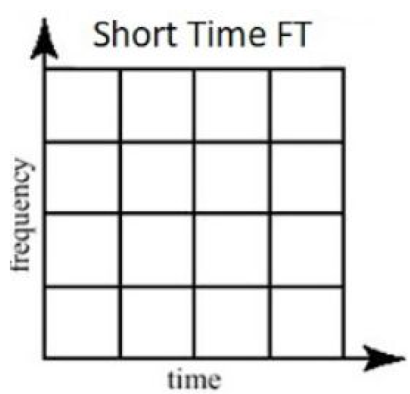
$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-\iota \omega t} dt$$

Fourier Transform to change the domain of function from t to w

In STFT (Short Time FT), we use the window function to add the time element. But there, the problem of the resolution arises.

$$F(\omega, \tau) = \int_{-\infty}^{\infty} f(t) \varphi(t - \tau) e^{-\iota \omega t} dt$$

STF Transform



- Window size is constant. Because of that, there is a wastage of precision of time-frequency values.
- Narrow Window gets us good time resolution, but poor frequency resolution and vice-versa.

Just like we represented a function as the combination of sine functions in FT, similarly in the WT, we represent the function as a combination of wavelets of different scale and position. A wavelet is a small wave function with average value 0. The great thing about WT is their multiresolution system.

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi(\frac{t-a}{b}) \quad a, b \ \epsilon \ \mathbb{R}$$

Wavelet function in time domain

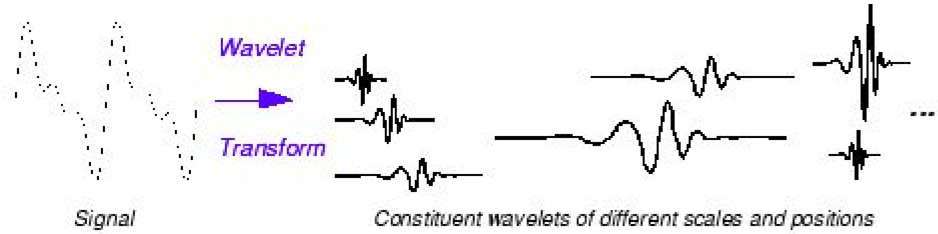
a is the dilation (scaling) parameter and **b** is the translation(position) parameter.

$\Psi(t)$ is also known as the mother function or prototype function.

$$\psi_{a,b}(\omega) = \sqrt{a}\psi(a\omega)e^{-\iota\omega}$$

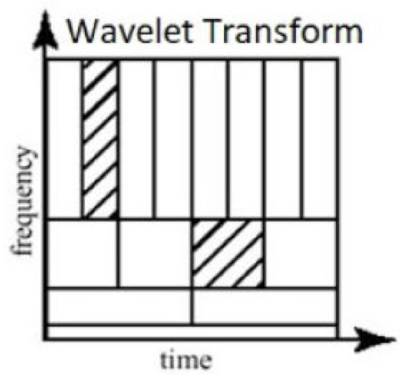
Wavelet function in the frequency domain

Basically, the wavelet transform is the projection of the raw function to the wavelet functions. I.e the Inner product of our function with various scaling and positions of the mother wavelet.



The window now varies according to the scale. For higher frequencies we narrow the window for better time resolution and for the lower frequencies we widen the window for better frequency resolution.

The assumption is that the low frequencies do not change that much with time, on the other hand the higher frequencies need the high time resolution for monitoring even a small change.



$$CWT(a,b) = < f, \psi_{a,b} > = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{t-b}{a})dt$$

Continuous Wavelet Transform

$< f, \psi_{a,b} >$ is the inner product.

CWT generates a function in the terms of a and b which are infinite, thus it is not plausible for practical use and thus we shift to the Discrete WT.

The way a,b are chosen give rise to two way of DWT that are :

- Redundant WT (Frames)
- MRA (Multi-Resolution Analysis)

Multiresolution analysis and Scaling

Need of Multiresolution: It is our common observation that the level of details within an image varies from location to location. Some locations contain significant details, where we require finer resolution for analysis and there are other locations, where a coarser resolution representation suffices. A multi-resolution representation of an image gives us a complete idea about the extent of the details existing at different locations from which we can choose our requirements of desired details. Multiresolution representation facilitates efficient compression by exploiting the redundancies across the resolutions. Wavelet transforms is one of the popular, but not the only approaches for multi-resolution image analysis. One can use any of the signal processing approaches to sub-band coding, such as Quadrature Mirror Filters (QMF) in multi-resolution analysis.

Scaling Functions and functional subspace: Any function f(x) can be analyzed as a linear combination of real-valued expansion functions $\phi_k(x)$.

$f(x) = \sum_k \alpha_k \phi_k$ ....(1)

where k is an integer index of summation (finite or infinite), the α_ks are the real valued expansion coefficients and {φ k (x)} forms an expansion set.

Let us compose a set of expansion functions { $\phi_{r,s}(x)$} through integer translations and binary scaling's of the real, square-integrable function φ(x), so that

$\phi_{r,s} = 2^{r/2}\,\phi(2^r x - s)$ ......(2)

where, $r, s \in Z$ (the integer space) and $\phi(x) \in L^2(R)$ (the square-integrable real space). In the above equation, s controls the translation in integer steps and r controls the amplitude, as well as the width of the function in the x-direction. Increasing r by one decreases the width by one-half and increases the amplitude by $\sqrt{2}$. In other words, the index r scales the function and the set of functions {$\phi_{r,s}(x)$} obtained through equation (2) are referred to as scaling functions. By a wise choice of φ(x) , the set of functions {$\phi_{r,s}(x)$} can be made to cover the entire square-integrable real space $L^2(R)$. Hence, if we choose any particular scale, say, $r = r_0$ the set of functions $\phi_{r,s}(x)$ obtained through integer translations can only cover a subspace of the entire $L^2(R)$. The subspace V_{r_0} so spanned is defined as the functional subspace of { $\phi_{r,s}(x)$} at a given scale r_0. Since the width of the set of functions {

$\phi_{r_0+1,s}(x)$ } is half of that of the set of functions { $\phi_{r_0,s}(x)$}, the latter can be analyzed by the former, but not the other way.

Multiresolution Properties :

Nesting: $.....V_{-1} \subset V_0 \subset V_1 \subset V_2.......$

Closure: closure ($\cup_{j\epsilon Z} V_j$) = $L^2(R)$

Shrinking: $\cap_{j\epsilon Z} V_j$ = {0}

Scaling: If f(t) ϵV_j Then f($2^{-j}t$) ϵV_0

Shift orthogonality: < $\phi(t), \phi(t-n)$ > =0 where n! =0.

Haar wavelet transform

Haar wavelet functions have been used in many fields, such as mathematics, sciences, biometric, and computer applications since 1910. Haar wavelets were introduced by a Hungarian mathematician named Alfred Haar. Haar worked in analysis studying orthogonal system functions, linear inequalities, and Chebyshev approximations. Haar defined the Haar wavelet theory in 1909, and this was the simplest of all wavelets. Mathematical representation of the Haar technique is termed as Haar wavelet transform.

The mathematical representation and evaluation of HWT is simple and can easily be implemented in terms of subtraction, addition, and division by 2. Like all wavelet transformations, the HWT decomposes the discrete time signal into two sub-signals of half its length also known as “details level.”

The wavelet family (φi,j(x))i∈N,j∈Z is an orthogonal subfamily of Hilbert space L2R implies that all functions in the wavelet family are obtained from a fixed function φ known as mother wavelet through translation and dilation. The wavelet family satisfies the following equations

ρi,j(x)=2j2 φ(2jx−i)

Wavelet Compression

The aim of the wavelet transformation function is not only to deal with the expectation of sparsity of the transformed matrices but also to enable us to shrink the “significant detail level” to set a lot of entries to zero. This helps in changing the transformed matrices entries taking the advantages of “regions of low activity,” and then the approximately original data can be reconstructed by applying the inverse wavelet transformation to the processed transformed matrix.So, we come up with the idea of using wavelet compression: define a threshold value (always positive) “α” and decide that every entry in the transformation matrix “T” that is less than or equal to “α” will be set to zero. Thresholding results in a sparse matrix. Then, by applying this transformation, we can reconstruct the original approximate data. We throw a sizable detail coefficient and satisfy ourselves with the result that is visually acceptable. The process will be a lossless compression if no information is lost and the original information is reconstructed properly (e.g. if $\alpha = 0$); otherwise, it will be a lossy compression in the case if it loses some information (if $\alpha > 0$)

$T(x) = \{0 : \text{if } |\alpha| \leq \varepsilon x : \text{otherwise}$

In the process of thresholding, a non-negative number known as threshold value has been selected, and any detail coefficient whose value is equal or less than the selected threshold value in the transpose matrix “T” will be set to zero. An optimum threshold value of 20 is selected in this case because if the selected threshold value is greater than the one that requires, then we will lose a lot of significant information, and as a result, the constructed image will be blurred more depending upon the threshold value selected. The result of different threshold values is shown in Figure 2.

Figure 2: (a) Threshold value=20, (b) threshold value=40, and (c) threshold value=10

A graphical representation for the same input image with multiple threshold values selected and its effect on the resultant image is shown in Figure 3.

From the graph shown in Figure 3, it is concluded that as we increase the threshold values, the compression rate increases, but on the other side, for high threshold values, the PSNR value decreases which results in low quality image. So, in this case, a threshold value of 20 is an optimal value for both PSNR value and compression rate.

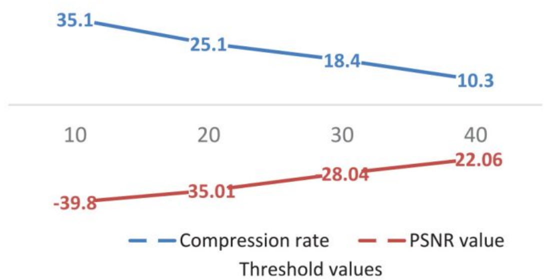


Figure 3: Threshold versus compression achieved.

A matrix with a high proportion of zero(s) is called a sparse. It is noted that the transformed matrix is sparser than the original matrix in lossy image compression. In our case, we set a threshold value of “20,” which means that we reset the values of details coefficients to zero whose values are less than or equal to 20 in the transformed matrix. The resultant matrix “D” is shown in Table 4.

	1212	−306	−146	−54	−24	−68	−40	0
	30	36	−	90	0	0	0	0
	−50	0	0	−24	0	72	0	0
	82	38	−24	68	48	−64	32	0
D =	0	0	−32	0	−48	−48	0	0
	0	0	−56	0	0	32	0	0
	0	0	−48	0	0	0	0	0
	44	36	0	0	80	0	0	0

Table 4: Thresholding

After applying the inverse transform to obtain the original approximate data, we get the resultant matrix “R” shown in Table 5.

R =	582	726	1146	1234	1344	1424	1540	1540
	742	694	1178	1074	1344	1424	1540	1540
	706	754	1206	1422	1492	1572	1592	1592
	818	866	1030	1374	1492	1572	1592	1592
	856	808	956	1220	1574	1590	1554	1554
	952	904	860	1124	1574	1590	1554	1554
	776	760	826	836	1294	1438	1610	1610
	456	760	668	676	1278	1422	1594	1594

Table 5: Reconstructed image.

Image compression techniques are used for compressing images with no loss of significant information. The results proved that the optimized HWT has high PSNR and more CR as compared to DCT and RLE. The proposed algorithm presented in this paper is more efficient, cost effective, simpler, and easy to implement. In addition, this algorithm helps in achieving high CR along with high PSNR value that promises significant details and high-quality during reconstruction of image. Sometimes, some image compression algorithms are hard to implement or lose significant information during image reconstruction, but the proposed method promises the reconstruction process with 98% accuracy with a minor loss of redundant information.

HARMONIC WAVELET TRANSFORM

Wavelet analysis is an important tool in image processing. In order to approximate or compress data, there are two common wavelet algorithms: the periodic wavelet algorithm, and the folded wavelet algorithm. More precisely, let an image f be supported on a square $\Omega \in R^2$. In the periodic wavelet algorithm, one extends f to a periodic function f^*, and then expands f^* into a periodic wavelet series using Daubechies wavelets or polyharmonic spline wavelets . Since f^* is discontinuous at the boundary points of Ω in general, the decay rate of the corresponding periodic wavelet coefficients is very slow. Hence, in order to obtain a good approximation of the image, we need many periodic wavelet coefficients. In the folded wavelet algorithm, in order to avoid the boundary mismatches caused by the brute-force periodization, one does an even extension of f, and then extends it to a periodic function f^*, and finally expands f^* into a periodic wavelet series with respect to a pair of biorthonormal periodic wavelet bases. If f is smooth, then $f^* \epsilon\, lip1\, on\, R^2$.This makes the decay rate of the corresponding wavelet coefficients faster than that in periodic wavelets algorithms. But, since $\frac{\mathrm{d}f^*}{\mathrm{d}x}\, and\, \frac{\mathrm{d}f^*}{\mathrm{d}y}$ are discontinuous at the boundary points, the decay rate of the periodic wavelet coefficients is still relatively low. Thus we use Harmonic wavelet Transform . In the HWT algorithm, the first step is to decompose $f = u + v$ and obtain a periodic function v^*,next a pair of real-valued biorthonormal wavelets ψ ,$\bar{\psi}$ of $L^2(R)$ generated by the real-valued even scaling functions ψ ,$\bar{\psi}$. Using a method of the tensor product and periodization, one gets a pair of biorthonormal periodic wavelet bases. Finally, v^* is expanded into the periodic wavelet series with respect to this pair of biorthonormal periodic wavelet bases. Since $\frac{\mathrm{d}v^*}{\mathrm{d}x}\, \epsilon\, lip1$ and $\frac{\mathrm{d}v^*}{\mathrm{d}y} \epsilon\, lip1$ on R^2 the decay rate of the corresponding periodic wavelet coefficients is faster than that of the periodic wavelet algorithm or the folded wavelet algorithm. In general, the decay rate of the Fourier sine coefficients depends on global smoothness, while that of the periodic wavelet coefficients depends on local smoothness. Since the global smoothness of a function is determined by its rough part, we need fewer periodic wavelet coefficients than the Fourier sine coefficients in order to approximate the image with the same quality. So, the HWT algorithm compresses data efficiently.

In the Harmonic Wavelet Transform algorithm, the symmetry of the periodic wavelet coefficients is carefully studied and show where these coefficients vanish. From this, we see that in order to recover the image exactly, the number of the efficient coefficients is just the same as the size of the sample points off. So he Harmonic Wavelet Transform algorithm is not a redundant transform.

The HWT algorithm is quite different from polyharmonic wavelets proposed by Van De Ville et ali. , which are a kind of wavelet bases constructed by polyharmonic functions. The HWT has a different strategy: approximate a target function by a harmonic function such that the residual part vanishes on the boundary, and then expand the residual part into wavelet series.

What is Singular Value Decomposition?

It states any matrix even rectangular A_{m*n} can be expressed as the product of two orthogonal matrices and a diagonal one. This factorization is expressed as

$A=U\Sigma V^T$, $U^TU=I_{m*m}$ and $V^TV=I_{n*n}$

where $U_{m\times m}$ and $V_{n\times n}$ are orthogonal matrices and $\Sigma_{m\times n}$ is a diagonal matrix whose diagonal entries are known as the singular values of the matrix A ,while columns or U and V are left singular vectors and right singular vectors respectively.The diagonal entries σ_i are basically square root of eigen values of A^TA. The number of non zero values in a diagonal matrix.

is the rank of A Its unique for a matrix if a non-decreasing order of singular values are chosen.

The Singular Value Decomposition (SVD) separates any matrix into m rank 1 pieces uv^T where r is the rank of matrix.

$A=\sigma_1u_1v^T_1+\cdots\cdots+ \sigma_ru_rv^T_r$

Where m is the rank of A and u_i, v_i represent the i-th columns of the matrices U and V respectively. Each piece is a column vector times a row vector scaled by a singular value and the The best k rank approximation to the fr obvious norm is given by

$A_k=\Sigma\sigma_iu_iv^T_i$ and $k < r$

Image Compression Using Svd

Each gray scale image can be viewed as m*n matrix where entries have intensity based on color scheme and encoding used. For simplicity 255 range is taken , where black(0) to white(255) and such matrix has rank r. As the singular values of decomposition satisfies the inequality $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_r>0$. This fact is used for compression. Instead of storing the whole expansion , only k terms are kept which also gives the best k rank approximation to the image.For colored image separate the image into three color matrix and apply the svd and k rank approximation and merge the result.

This means that the importance of the terms in the expansion decreases as more terms are considered, or equivalently, the first terms of the expansion must contain the most important information about the matrix A. This re-markable fact about the SVD of X is exactly what we can exploit to achieve compression: instead of storing the whole expansion we can try to store just a truncation of such expansion to k terms,The lower the value of k, the more size is reduced and compression is achieved but the quality also reduces.

References

1. Ahmet. “A guide for using wavelet transform.” http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/.
2. Sumeet. “The Wavelet Tutorial.” http://web.iitd.ac.in/~sumeet/WaveletTutorial.pdf.
3. “Wavelet Transform an Overview.” https://www.sciencedirect.com/topics/computer-science/wavelet-transforms.