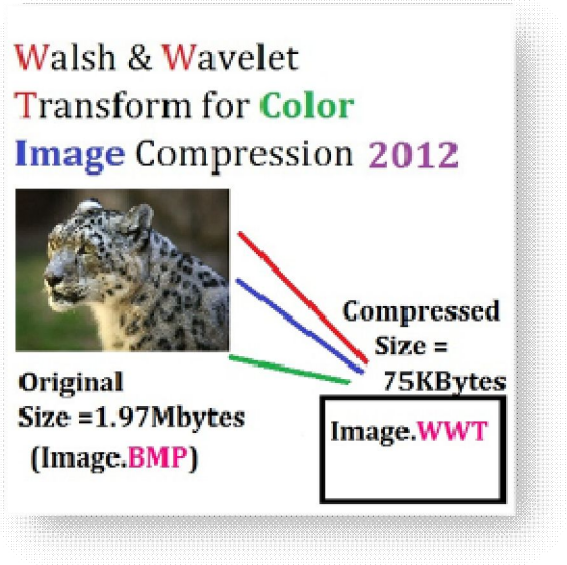
Color Image Compression by using Walsh and Wavelet Transform (WWT)

This Program designed
By
Mohammed Mustafa Siddeq
Lecturer and researcher
Software Engineering Dept.
Technical College – Kirkuk –IRAQ
Email: mamadmmx76@yahoo.com



Introduction Compression Algorithm (WWT)

This program is depends on Wavelet and Walsh transform for transformation, and then using Arithmetic coding for compress an image.

My Compression algorithm consists of the following steps:

A) Two Levels Discrete Wavelet Transform
B) Apply 2D Walsh-Hadamard Transform on each 8x8 block of the low-frequency sub-band
C) Split all DC values form each transformed block 8x8
D) Compress each sub-band by using Arithmetic coding

The following **Figure-1** shows first part of WWT compression algorithm steps for high-frequency domains, and then **Figure-2** shows second part of WWT compression steps for low-frequency domains.

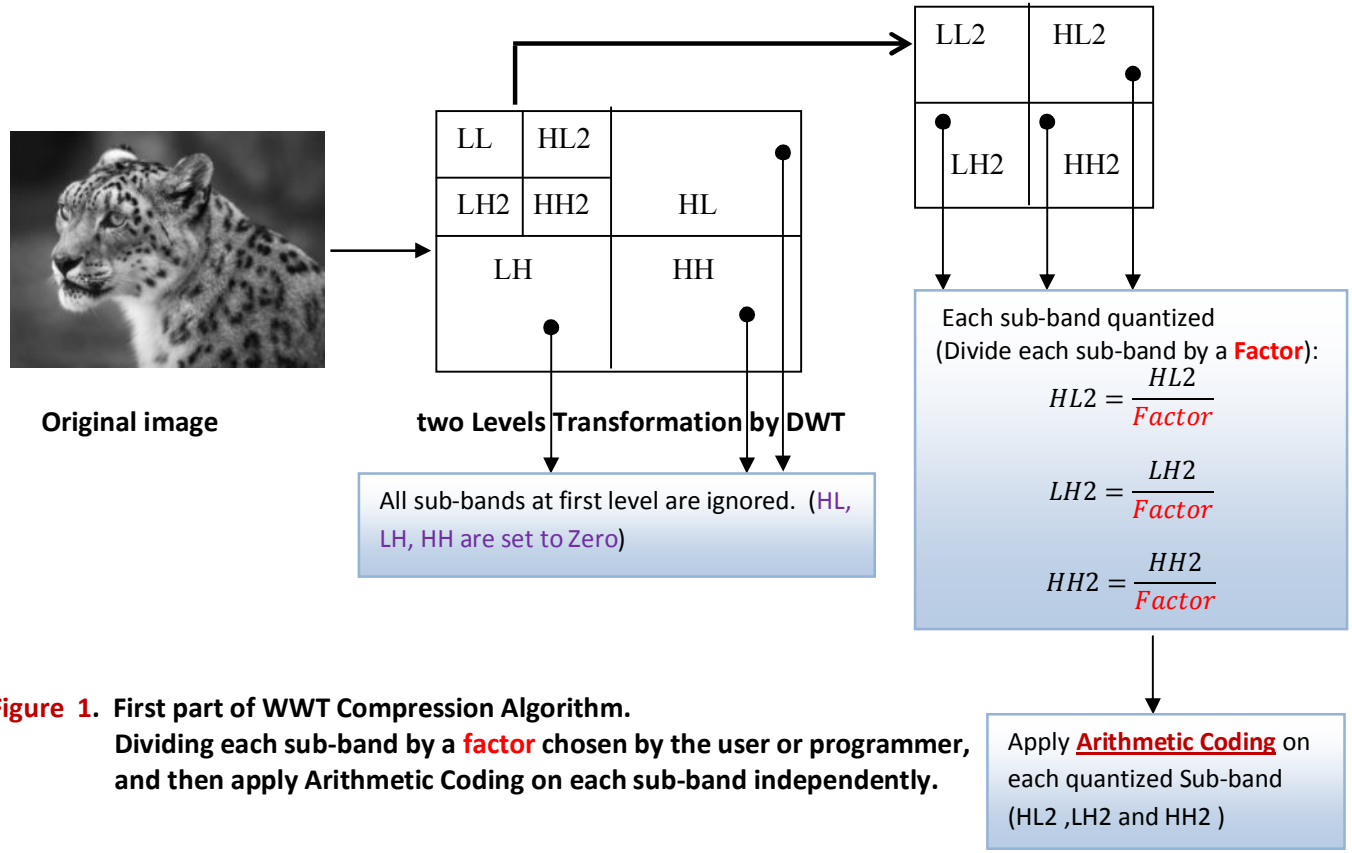


Figure 1. First part of WWT Compression Algorithm.
Dividing each sub-band by a factor chosen by the user or programmer, and then apply Arithmetic Coding on each sub-band independently.

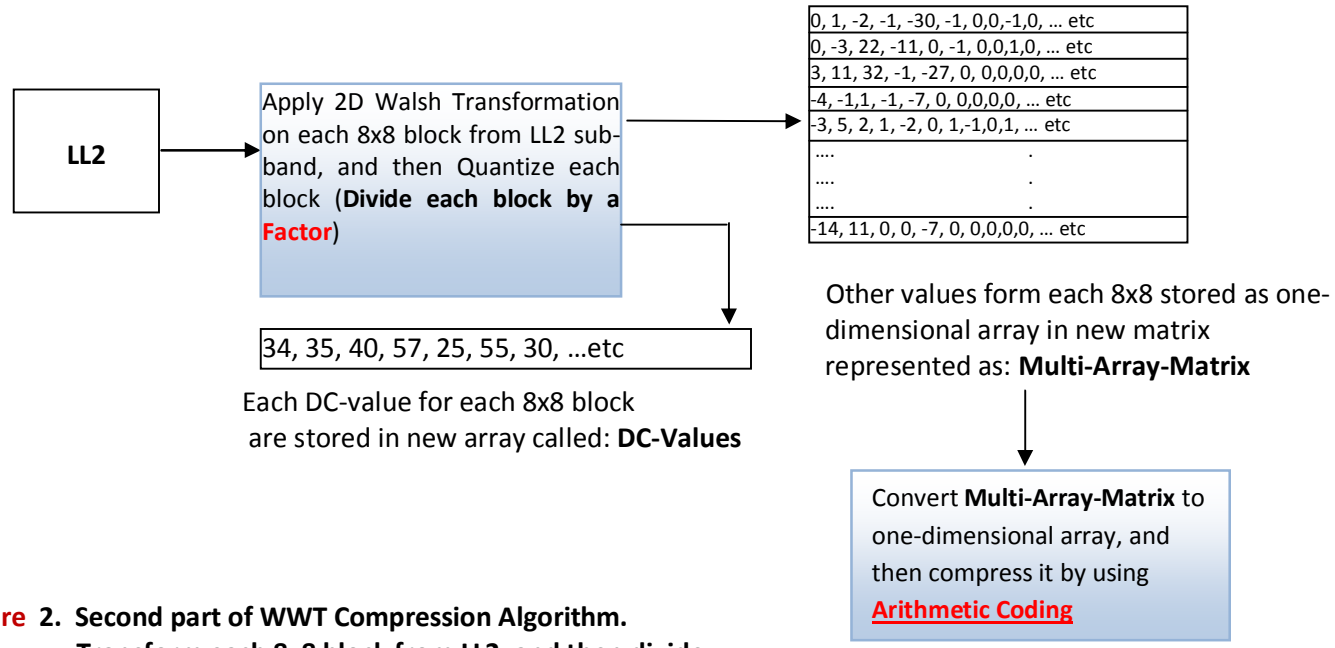


Figure 2. Second part of WWT Compression Algorithm.
Transform each 8x8 block from LL2, and then divide each block quantized by a factor chosen by a user/programmer. Each block 8x8 separated into; DC value and 63 coefficients, all DC values are stored in new array and saved in header-compressed file. 63 coefficients stored in new matrix, and compressed by Arithmetic coding.

Very important Note\ before applying this software please downloads the following functions:

1 – "**Arith_Code.m**" and "**Arith_Decode.m**"

2 – "**Walsh2D_Transform.m**"

These functions are free for download from WWW.MathWorks.Com ---> Atuher --> Siddeq

1. Program steps for compress gray scale images

In the first example, we learn how WWT compress a grayscale images:

```
% Example (1) --- for compression grayscale images-------------------------
    I = imread('D:\images\image1.bmp'); % read an image
    Factor1=0.025; % this factor for low-frequency sub-band (LL2)
    Factor2=0.025; % this factor for high-frequency sub-bands (HL2, LH2, and HH2)
    Para=2; % divide LL2 by this value for reduce LL2 values range
    [Header]=Walsh_DWT_Coding(I,[Factor1, Factor2],'db3', Para); % Apply compression
```

The compressed data in "`Header`", but it isn't in the disk…..
Using the following command in Mathlab language to store "`Header`" in the disk as the file name"`image1.WWT` " :

```
save ('D:\images\image1.wwt', 'Header'); % save compressed data in the file....
```

In the second example, we will change parameters:

```
% Example (2) ------for compression grayscale images-----------------------
    I = imread('D:\images\image1.bmp'); % read an image
    [Header]=Walsh_DWT_Coding(I,[0.05, 0.025],'db5',3); % Apply compression
   save('D:\images\image1.wwt', 'Header'); % save compressed data in the file....
```

Compression Parameters options:

- **A)** `'I'`: represented 2D image matrix with size **n x m**
- **B)** `'Factor1'` or `'Factor2'`: **-For Good image quality** : 0.02 or 0.03 or 0.04
- **C)** `'Factor1'` or `'Factor2'`: **-For Good compression ratio** : 0.05 or 0.07 or 0.1, …, 0.5
- **D)** `'dbN'` : using Daubechies wavelet transform filter or you could use another filter type
- **E)** `'Para'` : this factor is used for reduce low-frequency quality to Increase compression ratio.
  Examples: 2 or 3 or 4, ..., 10

2. Program steps for Decompress grayscale images

To test this software for compress and decompress using the following example without saving on the disk:

```
% Example (1) --- compression grayscale images without using Disk ---
 I = imread('D:\images\image1.bmp'); % read an image
 [Header]=Walsh_DWT_Coding(I,[0.025, 0.025],'db3',2); % for coding…
```

The coded data "Header" represented as a record, the user/programmer can use it later for another purpose maybe. The decompression image illustrated in the following:

```
%--- for Decompression grayscale images without disk---------------
 [Im]=Walsh_DWT_Decoding(Header); % for decoding….just using "Header"
 imshow(Im); % show final decoded image
```

For decoding image from disk using the following example:

```
% Example (2) --- compression grayscale images in Disk ---%
 I = imread('D:\images\imageG.bmp'); % read an image
 [Header]=Walsh_DWT_Coding(I,[0.05, 0.05],'db5',3); % Apply compression
 save('D:\images\imageG.wwt', 'Header'); % compressed data saved in a file....

%--- for Decompression grayscale images from disk---------------
X=load('D:\images\imageG.wwt', '-mat'); % load compressed data from the file....
H=X.Header; % read "Header" from loaded "X", and then put it in new variable "H"
Im=Walsh_DWT_Decoding(H); % for decoding "H";
imshow(Im); % show final decoded image
```

3. Compress and Decompress Color images

Before applying **WWT** technique on the color images, the RGB color images are converts into YCbCr form, and then applying WWT technique on each layer independently, this means each layer from YCbCr are compressed as a grayscale image. **Figure-3** shows **WWT** applied on each YCbCr layer.

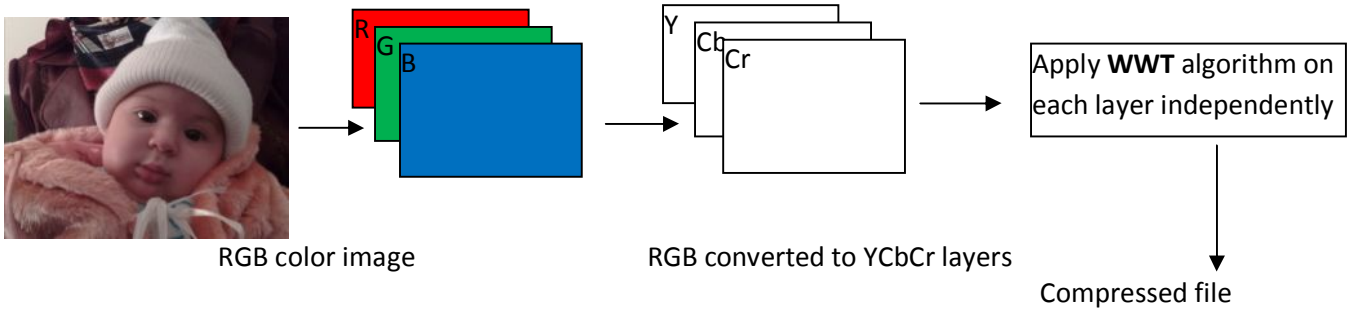


Figure -3. Shows RGB layers are converted to YCbCr layer, and the compressed by **WWT**

4. Program steps for Compression Color images

This program compress color image on the disk….

```
Path_Name='D:\lectures\images\color.bmp'; %assign file path and name
      Im = imread(Path_Name); % Read color image

      F1=0.05; % range values between {0.02 - 0.5} for low-frequency
      F2=0.05; % range values between {0.02 - 0.5} for high-frequency
      wave_name='db3'; % Wavelet filter used for decomposition
      Para=2; % range values {1 - 10}

      % ---- Convert Color image RGB to Ycbcr format
      ycbcr = rgb2ycbcr(Im); % this command is built-in at matlab language
      y=ycbcr(:,:,1); % put first layer in matrix "y"
      cb=ycbcr(:,:,2);% put second layer in matrix "cb"
      cr=ycbcr(:,:,3);% put the third layer in matrix "cr"

      % Apply compression WWT algorithm for each layer independently, and then put the
      %compressed data in records; "H1", "H2" and "H3" according to the layers sequences.
      H1=Walsh_DWT_Coding(y,[F1,F2],wave_name,Para);
      H2=Walsh_DWT_Coding(cb,[F1,F2],wave_name,Para);
      H3=Walsh_DWT_Coding(cr,[F1,F2],wave_name,Para);
```

```
%Store records "H1","H2" and "H3" in array of records "Header". This array combines
% all compressed data to be stored in one file…
Header(1).H1=H1;
Header(2).H2=H2;
Header(3).H3=H3;

%#########################################################################
% assign location for the compressed data as "*.wwt"
S_=size(Path_Name); % using same file path name
Path_Name(S_(2))='t'; Path_Name(S_(2)-1)='w'; Path_Name(S_(2)-2)='w';
%#########################################################################

save(Path_Name,'Header'); % Save Compress array of record "Header" in the disk
```

5. Program steps for Decompression Color images

This section used for read "Header" from the disk, and then apply decompression algorithm for show-decompressed image:

```
 Path_Name='D:\lectures\images\image7.wwt'; % assign compressed file
X=load(Path_Name, '-mat'); % Read all compress data from the file

Header=X.Header; % all compressed data in "X", and then read the array of recode "Header"

H1=Header(1).H1; % put the first compressed data in "H1"
H2=Header(2).H2; % put the second compressed data in "H2"
H3=Header(3).H3; % put the third compressed data in "H3"

% Apply Decompression on each layer
Y=Walsh_DWT_Decoding(H1); % decode first layer
Cb=Walsh_DWT_Decoding(H2);% decode second layer
Cr=Walsh_DWT_Decoding(H3);% decode third layer

%% Collect all layers in one matrix Ycbcr
ycbcr(:,:,1)=Y(:,:);
ycbcr(:,:,2)=Cb(:,:);
ycbcr(:,:,3)=Cr(:,:);
Im= ycbcr2rgb(ycbcr); % Convert YCbCr format to RGB color image

imshow(uint8(Im)); % show decompressed color image as "BMP"
```

Good luck for every programmer

Mohammed Mustafa Siddeq, He is one of the lecturer and researcher in Software Engineering Dept. – Technical College – Kirkuk – IRAQ. He is published many researches in different journals in UK and USA. He is professional in Mathlab language and Visual C++.NET. His interest area: Digital image processing, Neural Network, Genetic algorithm.