DCT Coefficients Compression Using Embedded Zerotree Algorithm

Dr. Tawfiq A. Abbas and Asa'ad N. Hashim

Abstract: The goal of compression algorithms is to gain best compression ratio with acceptable visual quality, the proposed compression satisfy this goal. Embedded Zerotree Algorithm works efficiently because of the following hypothesis:- " **If a DCT coefficient C at a coarse scale is insignificant with respect to a given threshold T, i.e. |C|<T then all DCT coefficients of the same orientation at finer scales are also likely to be insignificant with respect to T"**

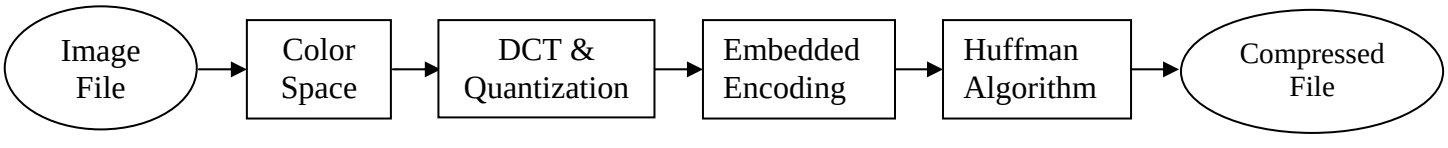
1.Introduction

Image compression is possible because images in general, are highly coherent(nonrandom),which means that there is redundant information. Visual data like other meaningful data, are usually structured, and this structure means that data over different parts of an image are interrelated. For example, consider an image in matrix format, if we take an arbitrary pixel, its color will likely be close to that of neighboring pixels, since they are more likely than not to belong to the same object. In any case, there are usually some redundant data because of the image structure. Image compression methods try to eliminate some of this redundancy to produce a more compact code that preserves the essential information contained in the image[2].

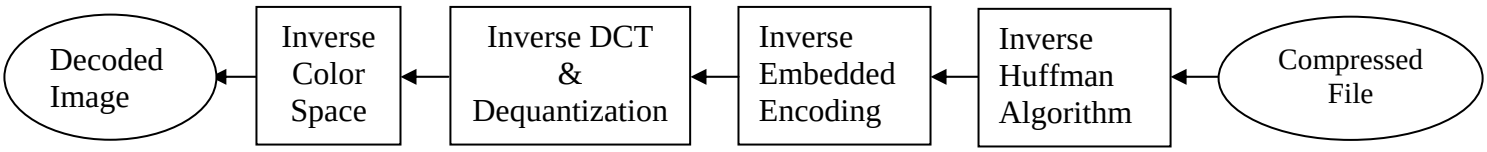
Main question about compression algorithms is how does one judge the quality of one versus another. In the case of lossless compression there are several criteria such that the time to compress, the time to reconstruct, the size of the compressed files, in the case of lossy compression the judgment is further complicated since we also have to worry about how good the lossy approximation is. There are typically tradeoffs between the amount of compression, the runtime, and the quality of the reconstruction. Depending on your application one might be more important than another and one would want to pick your algorithm appropriately[3].

2.The Proposed Compression System

This section presents practical implementation to general embedded bit streams for Discrete Cosine Transform(DCT) coefficients according to their importance. Two algorithms were suggested, the first one for compression and the other for decompression as shown in Figure(1) .



a. Compression

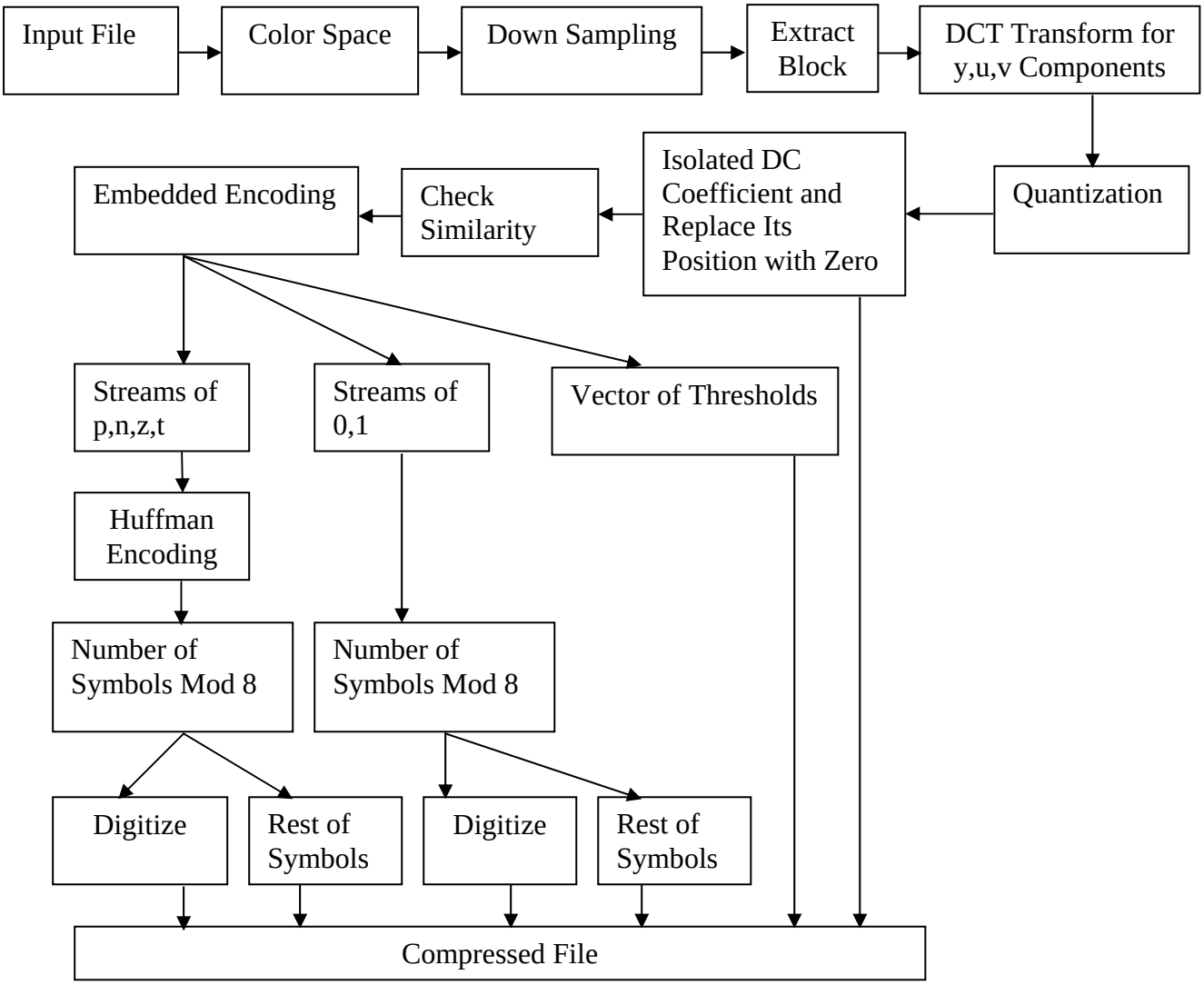


b. Decompression

Figure (1): Basic Components of The Proposed Compression System

2.1 The Proposed Compression Algorithm Steps

The stages of the suggested compression algorithm is illustrated in Figure (2) which represents block diagram for the Basic stages as well as the inputs, outputs , and the substages embedded in system.



Figure(2): Block Diagram for Basic Stages of The Proposed Algorithm

Embedded Zerotree Algorithm

Embedded Zerotree Algorithm presented by J. Shapiro(1993), consist of four main steps which are: threshold, dominant, subordinate, and loop. To make this algorithm appropriate to work with DCT coefficients which completely defers from Wavelet coefficients, the proposed system includes another steps to satisfy this goal. Figure(3) represents flowchart contain the main steps of Embedded Coding.

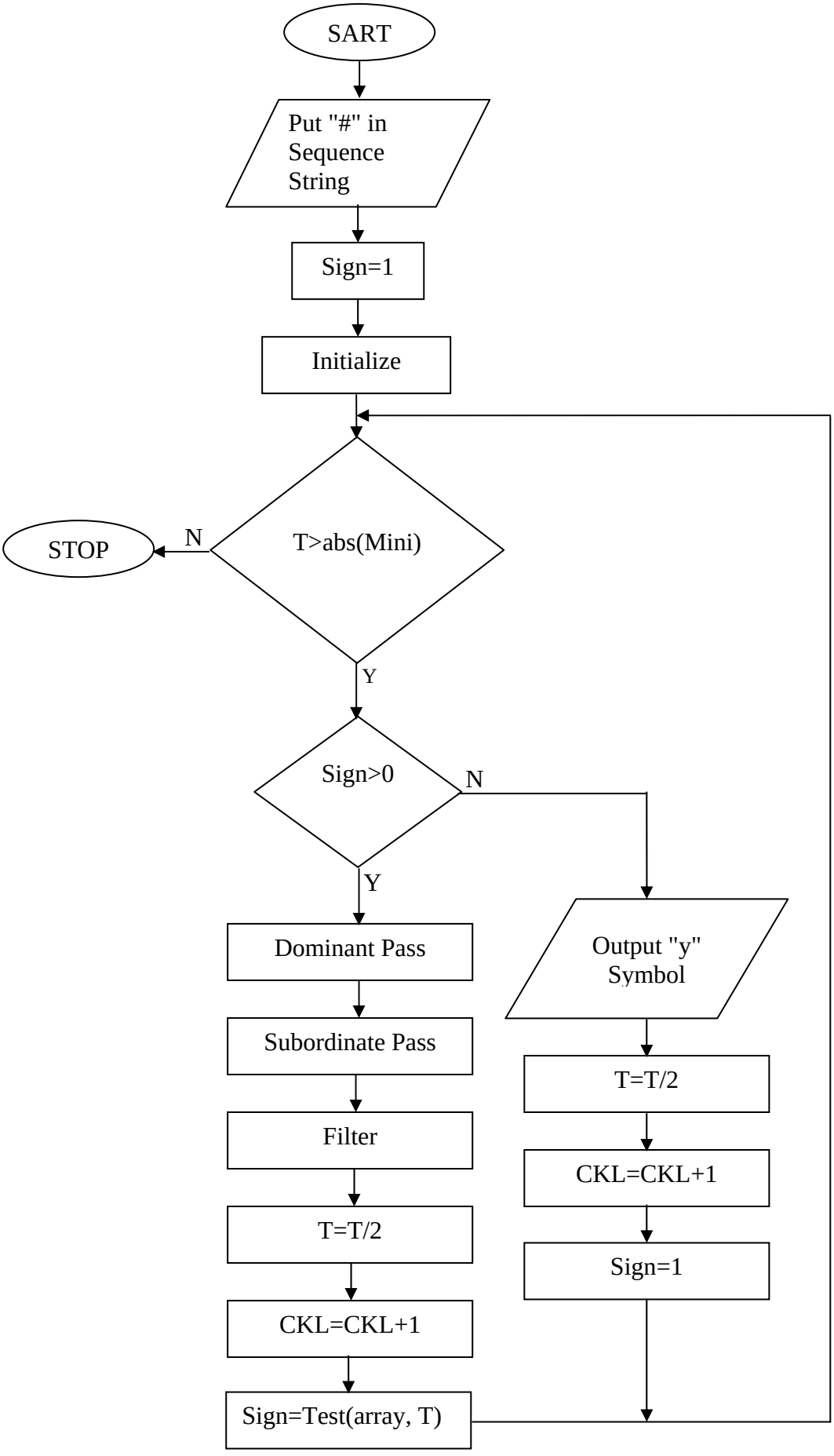
The used alphabetic contain eight symbols "p, n, z, t, #, @, -, y " , while first four symbols represent the original alphabetic used by Shapiro, other symbols added by proposed algorithm.

Where '#' represent an new block of coefficients,

'@' refers to similar block(all coefficients have same value),

'-' refers to a new row (in another word new cycle) and,

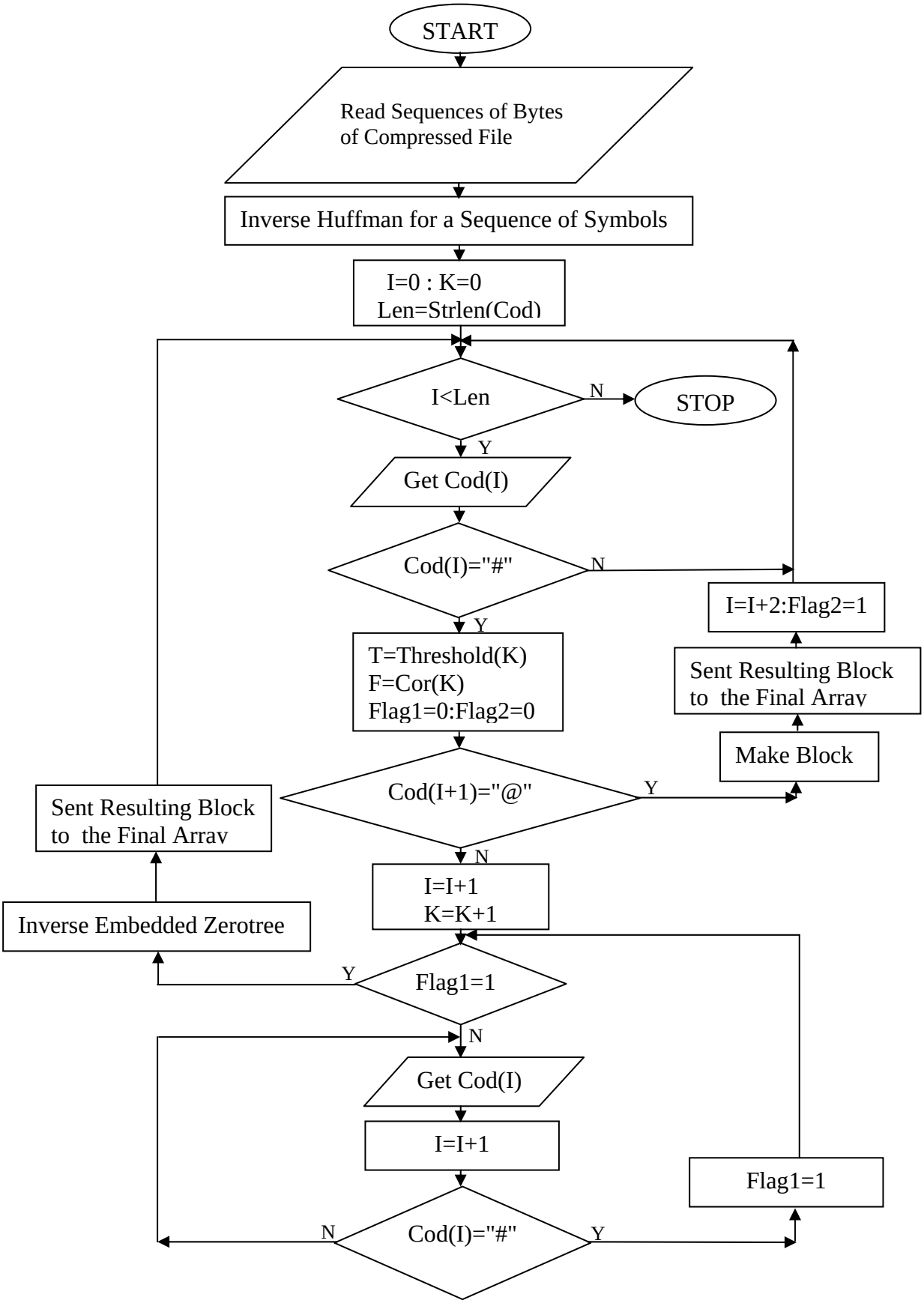
'y' refers to special case appear when the returned value(sign) from Test function equaled to zero.



Figure(3): Flowchart of The Embedded Zerotree Algorithm

2.2. The Proposed Decompression Algorithm Steps

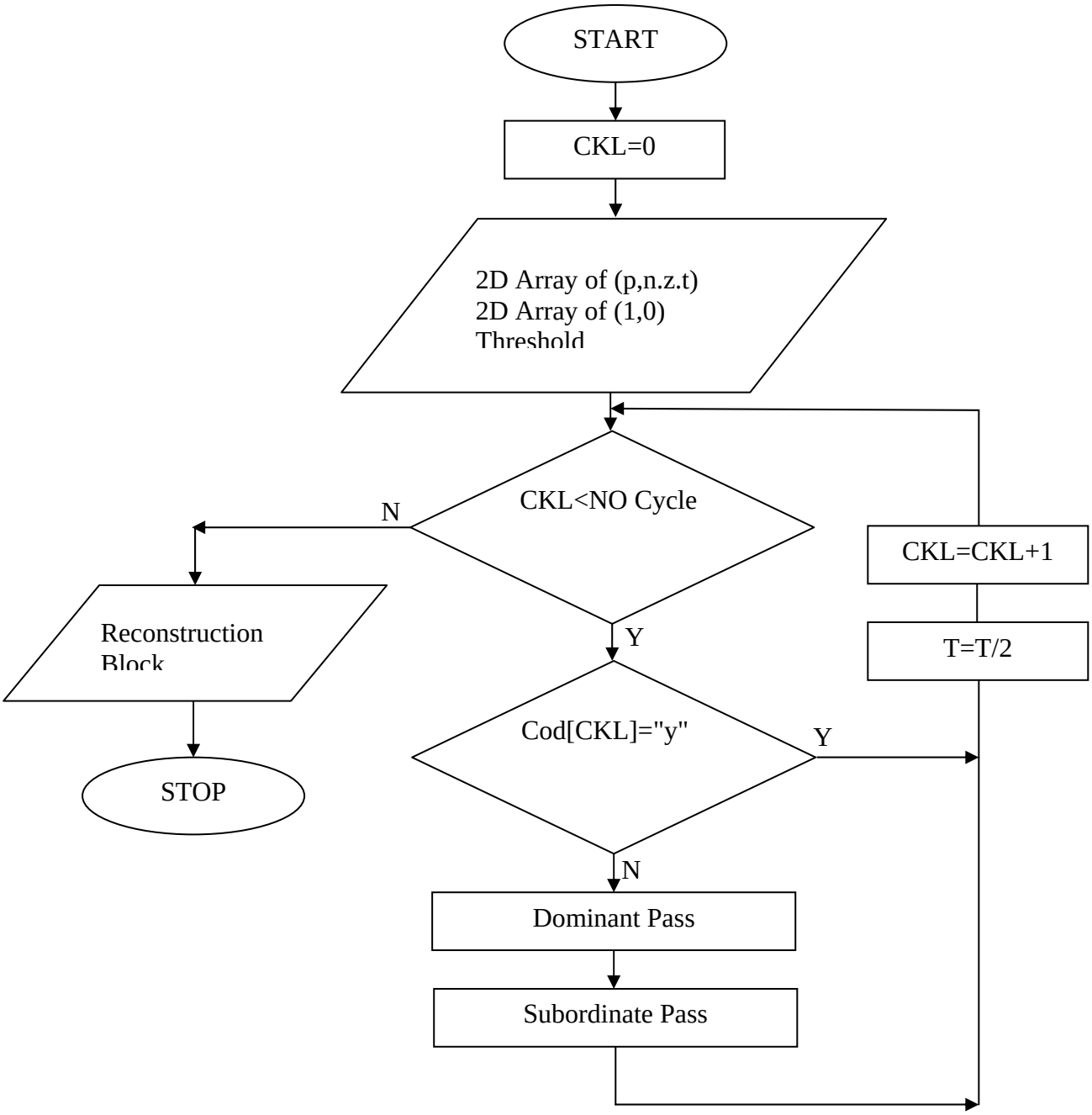
The steps of decompression algorithm are illustrated in Figure(4).



Figure(4): Flowchart of Decompression Algorithm

Inverse Embedded Zerotree Algorithm

The main stages of Inverse Embedded Coding is shown in Figure(5) which represents the flowchart of the basics operation.



Figure(5): Flowchart of Inverse Embedded Zerotree Algorithm

3. Characteristics of The Proposed Compression System Over EZW

- The proposed algorithm used eight symbols as alphabetic while **EZW** used just four symbols, in the first glance this is useless, but there a logical justification illustrated in the steps of compression/decompression algorithm.
- The proposed system treated main drop of **EZW** which is special case that shown in Figure (4.4), the treating was accomplished using **Test** Function.

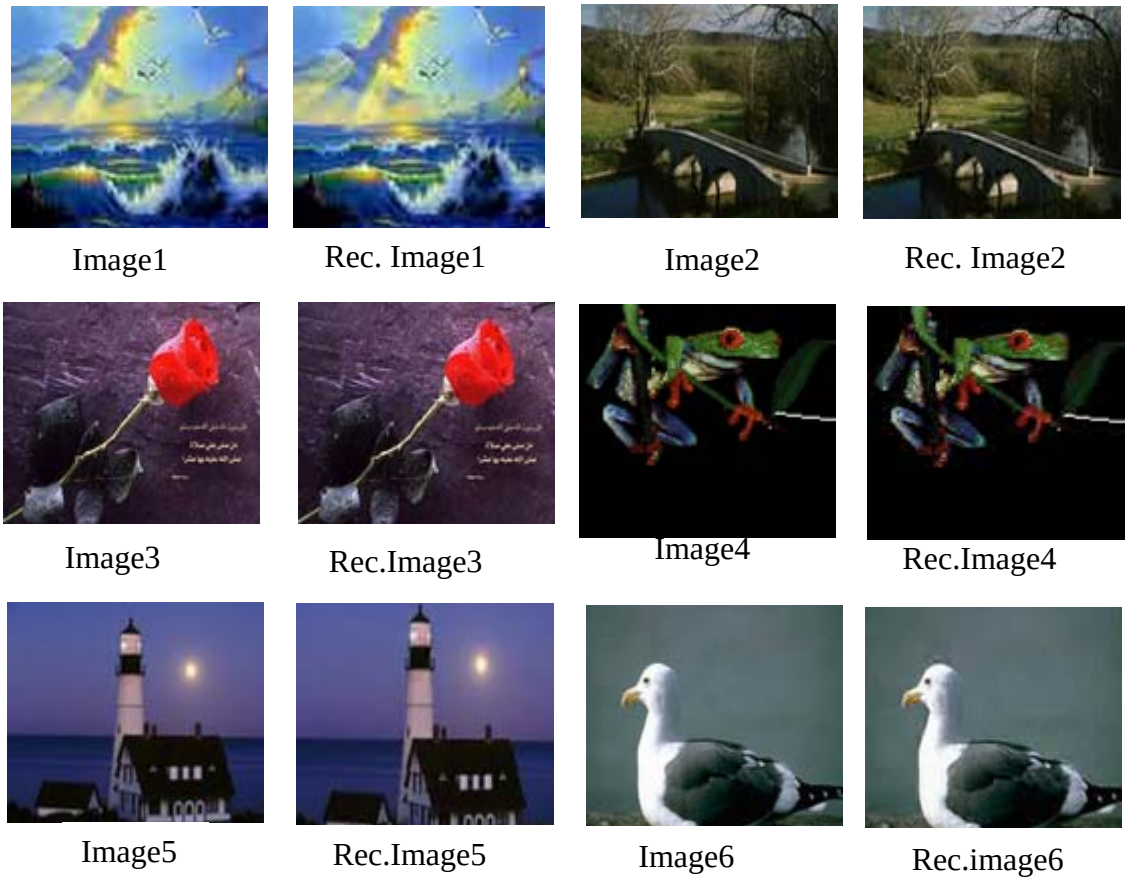
- **EZW** compress all blocks of data of image and don't care if the coming block has the same block (similarity case), this consume additional computation, time and resulting less compression ratio. In the suggested system there are specific symbol (@) to refers to this case.
- **Numbers** of cycle or level for each block derived from decompressed algorithm, there are not need for store this number in compressed file as **EZW** work.

4. Results of The Proposed Compression System

There are a number of images used to test the suggested compression system, this images illustrated in Figure(6) . The results of the proposed compression system are summarized in Table (1).

Table(1):Results of The Proposed Compression System

Source File Name	Source File Size	Compressed File Size	Compression Ratio	PSNR
Image1	49152(128*128)	4864	90%	33.6
Image2	98304(128*256)	7710	92%	27.77
Image3	98304(128*256)	7464.96	92%	27.15
Image4	49152(128*128)	3696.64	93%	28.96
Image5	196608(256*256)	11059.2	94%	34.58
Image6	98304(128*256)	6318.08	94%	31.7



Figure(6):Original and Reconstructed Images

5. Conclusions

In this section there are many conclusions:

1-The suggested system highlights the coding gained by DCT and Embedded Zerotree Algorithm, this allows to address the real issues involved in image coding, which are quantization and Entropy Coding.

2-Isolated DC's vector of blocks gives better visual quality and PSNR, with degrade the compression ratio.

3-Many researches considered EZW algorithm "does not really compression thing, it only reorder coefficients in such a way that can be compressed very efficiently"[8], while others showed "EZW algorithm which is considerably effective and important compression algorithm"[1]. The proposed system proved that Embedded Zerotree algorithm is effective compression algorithm.

4.The proposed system optimize the compression results by applying Huffman algorithm on the all result string instead of apply arithmetic algorithm for each block as in EZW

6. References

[1] Neyre T. and Hakan S., "**Embedded Zerotree Wavelet Compression**", Eastern Mediterranean University , 2005.

[2] Jonas G. and Luiz V., " **Image Processing for Computer Graphics**", Springer, 1997.

[3] Guy E., "**Introduction to Data Compression**", Carnegie Mellon University, 2001.

[4] Shapiro J., "**Embedded image using zerotrees of wavelet coefficient**", IEEE Trans. Signal Processing, vol. 41, pp3445-3462, Dec. 1993.

[5] Oliver E., Pascal F and Touradj E., "**Shape-Adaptive Wavelet Transform for Zerotree Coding**", In Proceedings of the Inter-national Conference on Acoustics, Speech, and Signal Processing ICASSP, Detroit, USA, May 1995.

[6] Said A. and Pearlman W., "**A new Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees**", IEEE Transactions on Circuits and Systems for Video Technology , Vol. 6, June 1996.

[7] Amir A., Moshe I. and Francois M., "**Speed Versus Quality in Low Bit-Rate Still Image Compression**", Signal Processing, 1999.

[8] Suresh S., "**Hardware Acceleration of the Embedded Zerotree Wavelet Algorithm**", East Tennessee State University, Master Thesis, 2004.